SECURING ROUTING PROTOCOLS IN MOBILE AD HOC NETWORKS

by

Mohamed Ahmed Abdelshafy Abdallah

Submitted for the degree of
Doctor of Philosophy

Department of Computer Science
School of Mathematical and Computer Sciences
Heriot-Watt University

May 2016

Abstract

A Mobile Ad Hoc Network (MANET) is more prone to security threats than other wired and wireless networks because of the distributed nature of the network. Conventional MANET routing protocols assume that all nodes cooperate without maliciously disrupting the operation of the protocol and do not provide defence against attackers. Blackhole and flooding attacks have a dramatic negative impact while grayhole and selfish attacks have a little negative impact on the performance of MANET routing protocols.

Malicious nodes or misbehaviour actions detection in the network is an important task to maintain the proper routing protocol operation. Current solutions cannot guarantee the true classification of nodes because the cooperative nature of the MANETs which leads to false exclusions of innocent nodes and/or good classification of malicious nodes. The thesis introduces a new concept of Self-Protocol Trustiness (SPT) to discover malicious nodes with a very high trustiness ratio of a node classification. Designing and implementing new mechanisms that can resist flooding and blackhole attacks which have high negative impacts on the performance of these reactive protocols is the main objective of the thesis. The design of these mechanisms is based on SPT concept to ensure the high trustiness ratio of node classification. In addition, they neither incorporate the use of cryptographic algorithms nor depend on routing packet formats which make these solutions robust and reliable, and simplify their implementations in different MANET reactive protocols.

Anti-Flooding (AF) mechanism is designed to resist flooding attacks which relies on locally applied timers and thresholds to classify nodes as malicious. Although AF mechanism succeeded in discovering malicious nodes within a small time, it has a number of thresholds that enable attacker to subvert the algorithm and cannot guarantee that the excluded nodes are genuine malicious nodes which was

the motivation to develop this algorithm. On the other hand, Flooding Attack Resisting Mechanism (FARM) is designed to close the security gaps and overcome the drawbacks of AF mechanism. It succeeded in detecting and excluding more than 80% of flooding nodes within the simulation time with a very high trustiness ratio.

Anti-Blackhole (AB) mechanism is designed to resist blackhole attacks and relies on a single threshold. The algorithm guarantees 100% exclusion of blackhole nodes and does not exclude any innocent node that may forward a reply packet. Although AB mechanism succeeded in discovering malicious nodes within a small time, the only suggested threshold enables an attacker to subvert the algorithm which was the motivation to develop it. On the other hand, Blackhole Resisting Mechanism (BRM) has the main advantages of AB mechanism while it is designed to close the security gaps and overcome the drawbacks of AB mechanism. It succeeded in detecting and excluding the vast majority of blackhole nodes within the simulation time.

To my Mother and Father

Acknowledgements

All thanks and praises to the Almighty Allah, the most Gracious and the most Merciful for his favour, grace, guidance and giving me the strength to achieve my PhD.

I would like to express my sincere thanks and greatest gratitude to my supervisor Dr. Peter King for his time, suggestions and continuous support during my research. You always help me to achieve my hopes and aspirations. I cannot find the words that you deserve. Many thanks Peter.

I would like to thank all staff members and administrative staff of Mathematical and Computer Science Department at Heriot-Watt University for their continuous help. My sincere thanks to Prof. Nicholas Taylor for his continuous support.

I am very grateful to the people I have been in contact with in Egypt, Saudi Arabia and UK. My gratitude to Heriot-Watt Muslim Society and Arab Society in Edinburgh for the nice time I have spent with them. Special Thanks to my office mate Turkey Alsalakini for his valuable support and help. My gratitude for my friends Ali Etorban, Idris Skloul and Mustafa Aswad for their valuable help. Thanks to my old and current office mates specially Khari Armih, Majed Al Saeed, Nawaf Mirza and Atif Al Ghamdi for their help and feedback.

Finally, I would like express my deepest thanks and greatest gratitude to my mother and father for their eternal favours and I would like to say to them without your prayers I did not achieve a success in my life and I would not be in this position. I would like to thank as well my wife, sons and daughter for their patience during my research. I would like to express my deepest gratitude to my brother and sisters. Thanks my family for your encouragement and support to finish my work successfully.

ACADEMIC REGISTRY
Research Thesis Submission

HERIOT WATT UNIVERSITY

Name:	**Mohamed Ahmed Abdelshafy Abdallah**		
School/PGI:	**Mathematics and Computer Sciences**		
Version: *(i.e. First, Resubmission, Final)*	**Final**	Degree Sought (Award **and** Subject area)	**PhD Computer Science**

Declaration

In accordance with the appropriate regulations I hereby submit my thesis and I declare that:

1) the thesis embodies the results of my own work and has been composed by myself
2) where appropriate, I have made acknowledgement of the work of others and have made reference to work carried out in collaboration with other persons
3) the thesis is the correct version of the thesis for submission and is the same version as any electronic versions submitted*.
4) my thesis for the award referred to, deposited in the Heriot-Watt University Library, should be made available for loan or photocopying and be available via the Institutional Repository, subject to such conditions as the Librarian may require
5) I understand that as a student of the University I am required to abide by the Regulations of the University and to conform to its discipline.

* *Please note that it is the responsibility of the candidate to ensure that the correct version of the thesis is submitted.*

Signature of Candidate:	MAbdelshafy	Date:	5-5-2016

Submission

Submitted By *(name in capitals):*	**Mohamed Ahmed Abdelshafy Abdallah**
Signature of Individual Submitting:	MAbdelshafy
Date Submitted:	5-5-2016

For Completion in the Student Service Centre (SSC)

Received in the SSC by *(name in capitals):*			
Method of Submission *(Handed in to SSC; posted through internal/external mail):*			
*E-thesis Submitted (**mandatory for final theses**)*			
Signature:		Date:	

Please note this form should bound into the submitted thesis.

Updated February 2008, November 2008, February 2009, January 2011

Contents

List of Tables

List of Algorithms

List of Figures

Glossary

AB	Anti-Blackhole
ABR	Associativity-Based Routing
AES	Advanced Encryption Standard
AF	Anti-Flooding
AODV	Ad hoc On Demand Distance Vector
AOMDV	Ad hoc On-demand Multipath Distance Vector Routing
ARAN	Authenticated Routing for Ad-hoc Networks
BRM	Blackhole Resisting Mechanism
CEDAR	Core Extraction Distributed Ad hoc Routing Protocol
CONFIDANT	Cooperation of Nodes Fairness in Dynamic Ad hoc Networks
DES	Data Encryption Standard
DoS	Denial of Service
DSA	Digital Signature Algorithm
DSDV	Destination Sequenced Distance-Vector
DSR	Dynamic Source Routing
ECC	Elliptic Curve Cryptography
FAP	Flooding Attack Prevention
FARM	Flooding Attack Resisting Mechanism
FSR	Fisheye State Routing
MANET	Mobile Ad Hoc Network
OLSR	Optimized Link State Routing
RERR	Route Error Packet
RREP	Route Reply Packet

RREQ	Route Request Packet
RSA	Rivest-Shamir-Adleman
SAODV	Secure Ad-hoc On-demand Distance Vector Routing Protocol
SAR	Security-Aware ad-hoc Routing
SEAD	Secure Efficient Ad hoc Networks
SHA	Secure Hash Algorithm
SLSP	Secure Link State Routing Protocol
SMORT	Scalable Multipath On-demand Routing
SPT	Self-Protocol Trustiness
SRP	Secure Routing Protocol
WRP	Wireless Routing Protocol
ZRP	Zone Routing Protocol

Publications

1. Mohamed A. Abdelshafy and Peter J.B. King, AODV Routing Protocol Performance Analysis under MANET Attacks, International Journal for Information Security Research (IJISR), Volume 3, Issues 1/2, Mar/Jun 2013, pages 418-426.

2. Mohamed A. Abdelshafy and Peter J.B. King, Analysis of security attacks on AODV routing, In 8th International Conference for Internet Technology and Secured Transactions (ICITST), pages 290-295, London, UK, Dec 2013.

3. Mohamed A. Abdelshafy and Peter J.B. King, AODV & SAODV under attack: Performance comparison, In ADHOC-NOW 2014, LNCS 8487, pages 318-331, Benidorm, Spain, Jun 2014.

4. Mohamed A. Abdelshafy and Peter J.B. King, AF-AODV: Mitigating the Impact of Flooding Attack on AODV, In 8th International Conference on Emerging Security Information, Systems and Technologies (SECURWARE), pages 1-6, Lisbon, Portugal, Nov 2014.

5. Mohamed A. Abdelshafy and Peter J.B. King, Dynamic Source Routing under attacks, In 7th International Workshop on Reliable Networks Design and Modelling, pages 174-180, Munich, Germany, Oct 2015.

6. Mohamed A. Abdelshafy and Peter J.B. King, Resisting Blackhole Attacks on MANETs, In 13th IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, USA, Jan 2016.

7. Mohamed A. Abdelshafy and Peter J.B. King, Flooding Attacks Resisting Mechanism in MANETs (Under Review).

Chapter 1

Introduction

Mobile Ad hoc Networks (MANETs) play an essential role in wireless communications research due to their infrastructure-less and characteristics. MANETs are used in many applications such as military, emergency and rescue operations and sensor networks. Routing protocols, security, medium access scheme, energy management, quality of service and scalability are major challenges that need to be considered in the design of MANETs. Mobility is the major issue that causes frequent link failures in MANETs, resulting in performance failure especially in situations of high node mobility. MANET nodes act as nodes and routers as well although their limited resources which imposes a burden to these nodes. MANETs are more prone to security threats than other wired and wireless networks because of the lack of any infrastructure.

1.1 Thesis Scope

Designing a secured routing protocol in MANETs is challenging because of the limitations and characteristics of the network. Current MANET routing protocols are designed based on an absence of any centralized authority and an assumption that nodes cooperate without maliciously disrupting the routing protocol. MANETs are more vulnerable to security attacks than other wired and wireless networks due to their inherent characteristics such as the wireless nature, node mobility, an absence of a centralized authority or trusted third party, and limited resources that impose a

difficulty to implement cryptography algorithms. So, MANETs have many security challenges that motivate researchers to resist different types of attacks from passive eavesdropping to active interfering.

1.2 Research Motivation

Reactive MANET routing protocols are vulnerable to a dramatic collapse of network performance under different attacks. Security solutions for wired networks cannot be applied directly to MANETs because of the nature of the network and because of an absence of a centralized authority. Current solutions can be mainly classified as prevention mechanisms, and detection and reaction mechanisms. Prevention mechanisms are designed based on the cryptographic algorithms which do not suit the MANET characteristics. In addition, these solutions succeed in discovering some attacks and fail in others especially the flooding attacks. On the other hand, detection and reaction mechanisms are used to discover malicious nodes or misbehaviour actions in the network to maintain the proper routing protocol operation. These solutions cannot guarantee the true classification of nodes because the cooperative nature of the MANETs which leads to false exclusion of innocent nodes and/or good classification of malicious nodes. Moreover, many solutions of both types are designed to suit a specific MANET routing protocol, as these solutions depend on the routing packet format of this protocol, which decreases the opportunity to use these solutions in other MANET routing protocols.

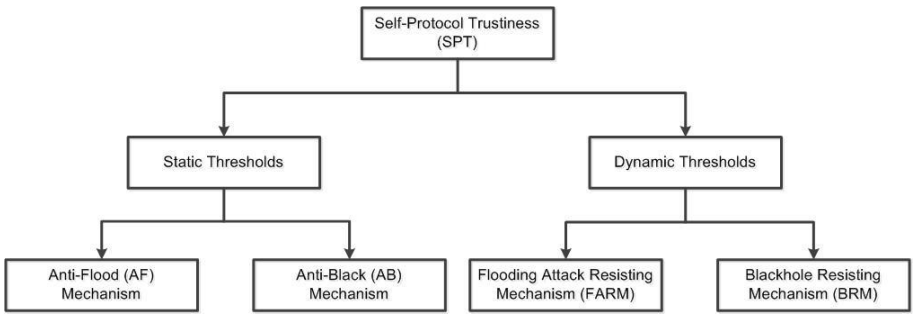

Figure 1.1: SPT and Mechanisms Relationship

This thesis introduces a new concept of Self-Protocol Trustiness (SPT) to discover malicious nodes with a very high trustiness ratio of a node classification. This

is because detecting a malicious intruder is based on an implied avowal of its malicious behaviour. SPT-based solutions do not require expensive cryptography or authentication mechanisms, or modifications to the packet formats of the underlying protocol which make these solutions able to be implemented in different MANET reactive protocols. SPT is used an underlying concept in designing mechanisms to resist both flooding and blackhole attacks as they have the most dramatic effect of the network performance. Figure 1.1 shows the relationship between the SPT concept and different solutions. Objectives of the thesis are:

1. Study and investigate the impacts of some attacks on the MANET reactive routing protocols that represent different types of attacks. Flooding, blackhole, grayhole and selfish attacks are selected to represent modification, impersonation and fabrication of the routing packets.
2. Designing and implementing new mechanisms that can resist flooding and blackhole attacks which have the high negative impacts on the performance of these reactive protocols. The design of these mechanisms should be based on SPT concept to ensure the high trustiness ratio of node classification. In addition, it should not incorporate the use of cryptographic algorithms to suit the limited resources of the MANET nodes and it should not depend on routing packet formats to allow implementation in different MANET reactive protocols.

1.3 Thesis Contributions

The thesis has the following contributions:

1. Analysis and evaluation of the performance of AODV, DSR, AOMDV and SAODV routing protocols under the blackhole, grayhole, selfish and flooding attacks. These analyses concluded that the blackhole and flooding attacks have dramatic negative impact while grayhole and selfish attacks have little negative impact on the performance of MANET routing protocols. The thesis concludes also that the highest negative impact of malicious nodes

in all these different attacks usually appears on static networks and this effect decreases as node mobility increases. The thesis concludes as well that AOMDV achieves a higher resistance to different attacks than other reactive protocols. While SAODV succeeded in resisting blackhole, grayhole and selfish attacks, it suffered performance degradation under the flooding attack. On the other hand, while SAODV achieves a moderate performance compared to the other protocols, its routing overhead is higher due to the cost of its security features.

2. Introducing a new concept in securing multi-hop networks such as MANETs called Self-Protocol Trustiness (SPT). This concept clarifies that a detection of a malicious intruder is accomplished by complying with the normal protocol behaviour and luring the malicious node to give an implicit avowal of its malicious behaviour.

3. Designing and implementing an Anti-Flooding (AF) mechanism to resist flooding attacks which can be incorporated into any reactive routing protocol. It does not require expensive cryptography or authentication mechanisms, but relies on locally applied timers and thresholds to classify nodes as malicious. No modifications are made to the packet formats and hence do not incur any additional overhead. The proposed mechanism succeeded in discovering malicious nodes that attempt to flood the network regardless of the number of malicious nodes.

4. Designing and implementing the Flooding Attack Resisting Mechanism (FARM) to resist flooding attacks. It has the main advantages of AF mechanism while it is designed to close the security gaps and overcome the drawbacks of AF mechanism. FARM mechanism uses our new concept of Self-Protocol Trustiness (SPT) to discover malicious intruders. It succeeded in detecting and excluding a high ratio of flooding nodes in a short time. FARM succeeded in detecting and excluding more than 80% of malicious neighbours in the simulation time.

5. Designing and implementing the Anti-Blackhole (AB) mechanism to resist blackhole attacks which can be incorporated into any reactive routing protocol. It is designed based on our new concept of Self-Protocol Trustiness (SPT) in addition to some thresholds to discover a blackhole node. The algorithm neither adds new routing packets nor modifies the existing ones and it does not require expensive cryptography or authentication mechanisms. The algorithm guarantees 100% exclusion of only blackhole nodes and does not exclude any victim node that may forward a reply packet.

6. Designing and implementing the Blackhole Resisting Mechanism (BRM) to resist blackhole attacks. It has the main advantages of AB mechanism while it is designed to close the security gaps and overcome the drawbacks of AB mechanism that enable malicious nodes to subvert it. It succeeded in detecting and excluding a high ratio of blackhole nodes in a short time.

1.4 Organisation of the thesis

The rest of this thesis is organised as follows:

Chapter 2: Introduces the literature overview of the MANET and its characteristics and discusses MANET routing protocols and their classifications. Then, it presents MANET routing attacks, a brief discussion about cryptographic algorithms and existing MANET secured routing protocols. Finally, systems modelling techniques are discussed.

Chapter 3: Introduces the research environment of the thesis. It presents the details about the selected MANET routing protocols, selected MANET routing attacks and the selected simulation tool. It discusses as well the reasons for selection for each item.

Chapter 4: Studies the behaviour of different routing protocols under various attacks. It analyses the behaviour of AODV, DSR, AOMDV and SAODV in the presence of flooding, blackhole, grayhole and selfish attacks under different

node mobility. Then, it outlines the comparison between these protocols in static networks.

Chapter 5: Discusses resisting the flooding attacks in MANETs. It begins with an overview of the existing solutions for these attacks mentioning the drawbacks of them. Then, it introduces our first solution, the Anti-Flooding (AF) mechanism to detect the flooding attacks that use some thresholds and timers to classify nodes as malicious. Later, it introduces our second solution to detect the flooding attacks which is called the Flooding Attack Resisting Mechanism (FARM) that is designed to close the security gaps and overcome the drawbacks of AF mechanism. FARM uses our new concept in discovering malicious nodes called Self-Protocol Trustiness (SPT). Finally, a simulation approach, parameters and results of simulating our solutions and comparing the results to the existing one are presented.

Chapter 6: Discusses resisting the blackhole attacks in MANETs. It begins with an overview of the existing solutions for these attacks discussing their drawbacks. Then, it introduces our first solution Anti-Blackhole (AB) mechanism to detect the blackhole attacks that is designed using SPT concept. Later, it introduces our second solution to detect the blackhole attacks which is called the Blackhole Resisting Mechanism (BRM) that is designed to close the security gaps and overcome the drawbacks of AB mechanism. Finally, a simulation approach, parameters and results of simulating our solutions and comparing the results with the existing solutions are presented.

Chapter 7: Provides a summary of the thesis, highlights the contributions and suggests future work.

Chapter 2

Literature Review

2.1 Introduction

Mobile Ad hoc Networks (MANETs) have been an important research area due to its infrastructure-less, self-configuration and self-maintenance characteristics [1] throughout the last two decades. Application domains include military operations, emergency and rescue operations, wireless mesh and sensor networks and collaborative and distributed computing [2]. Routing protocols, security, medium access scheme, energy management, quality of service and scalability are major challenges that need to be considered when a MANET is designed.

A number of routing protocols [3] for MANET have been proposed over the past years. Routing protocols exchange routing information such as hop count, sequence number, signal strength, and geographical information, and establish an efficient and feasible route to a destination node. The major issues involved in designing a routing protocol for ad hoc wireless networks are node mobility, bandwidth constrained wireless channel, resource constrained nodes, and error prone shared broadcast wireless channel. MANET routing protocols cope well with dynamically changing topology but are not designed to provide defence against malicious attackers [2]. Current routing protocols are not able to thwart common security threats. Most of these protocols do not incorporate any security and are highly vulnerable to attacks.

MANET is highly exposed to security attacks in comparison to traditional wired networks. There are a large number of existing attacks against MANET routing

protocols [4]. Malicious nodes and attackers inject erroneous routing information, replaying old routing information or distort routing information in order to partition a network or overload a network with retransmissions, thereby causing congestion, and hence a Denial of Service (DoS) attack can be launched. Detection of these malicious nodes through routing information is difficult due to the dynamic topology of MANET. Lack of central authority, shared broadcast channel, limited bandwidth and limited resource availability are some unique characteristics of MANET that cause difficulty in the designing of secured routing protocol.

Secure communication is very important in applications like military environments. Recently, several research efforts have been introduced to counter against malicious attacks on MANETs [2]. Most of the previous research has focused mainly on providing preventive schemes to protect the routing protocol in MANET. These schemes are designed based on key management or encryption techniques to prevent unauthorized nodes from joining the network and to discover misbehaviours. In general, the main drawback of these approaches is that they introduce a heavy traffic load to exchange and verify keys which is very expensive in terms of the bandwidth constraint for MANET nodes that have limited battery and computational capabilities.

The rest of the chapter is organized as follows. Section 2.2 presents an introduction to Mobile Ad Hoc Networks (MANETs). In Section 2.3, Mobile Ad Hoc Network characteristics are presented. Section 2.4 MANET routing protocols and their classifications are discussed. In Section 2.5, MANET routing attacks are presented. Section 2.6 introduces a brief discussion about cryptographic algorithms and existing secured MANET routing protocols. In Section 2.7, a summary is presented.

2.2 Mobile Ad hoc Networks (MANETs)

Wireless networks can be classified into two types; Infrastructure or Infrastructure less. In an infrastructure wireless network, devices such as access points and base stations are located throughout the network. Mobile nodes can maintain their connection with the network by disconnecting from the range of a base station and con-

necting to another base station. On the other hand, in infrastructure-less networks devices are connected without any fixed base stations. Mobile nodes dynamically establish and maintain routing among themselves as nodes act as routers. Figure 2.1 [5] presents an example of Mobile Ad Hoc Network.

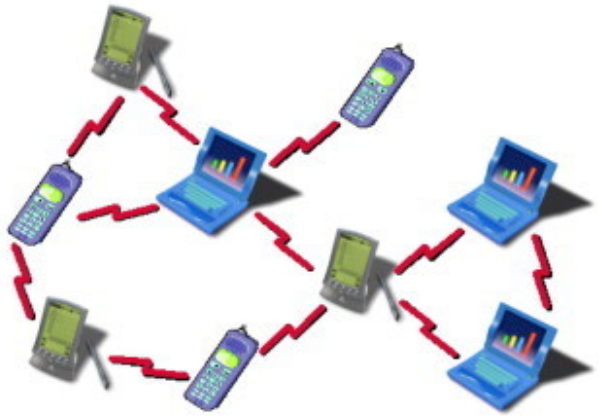

Figure 2.1: Mobile Ad hoc Network [5]

A MANET is a decentralized network in which there is no infrastructure to manage the information traffic between existing nodes. MANET can be defined as a collection of wireless mobile nodes that communicate with each other without centralized control or established infrastructures [6]. In a MANET, a wireless device (node) communicates directly with other wireless device inside their radio range in a peer-to-peer nature. If a source node needs to communicate with a destination node which is outside the source's range, it has to use intermediate nodes within its radio range to forward data to the destination. Each node in a MANET acts as a router and as a host. Node mobility is one of main features of MANET which causes network topology to change that requires frequent updating of routing information. In addition, MANET nodes are characterised by their limited resources such as power and computational capabilities.

2.3 MANET Characteristics

MANETs have become a major field of research in recent years because of the challenging problems they pose. The links in a MANET are more susceptible to transmission errors than wired links [6]. In addition, the mobility of the nodes leads

to links breaking and new links being formed as the nodes move in and out of range. MANETs have many challenges that impact on its performance, and as we focus in the thesis on MANET security, we can notice the impact of these challenges on its security in the following sections. These challenges can be summarized as follows:

2.3.1 Distributed Operation

Nodes in MANET have to cooperate with each other acting as a relay to implement functions such as routing and security. The lack of centralized management makes conventional trust management for nodes impossible and makes the detection of attacks difficult especially in a highly dynamic and large scale MANET [6].

2.3.2 Dynamic Topologies

Since the nodes are mobile, the network topology is susceptible to rapid and unpredictable changes. Mobility is one of the most important characteristics of MANETs which leads to a dynamic network topology. Mobility is the major issue that causes frequent link failures in MANETs, resulting in performance failure especially in high node mobility [3]. Dynamic topology forces nodes to continuously update their routing tables and share this routing information among them. Network topology changes as well may disturb any trust relationship among nodes especially if some nodes have been detected as malicious nodes.

2.3.3 Node-Constrained Resources

MANET nodes are mobile devices with limited CPU processing capability, small memory size, bandwidth and low power storage [2]. Therefore algorithms need to be optimized to minimize resource consumption which imposes a difficulty to implement cryptographic algorithms [4]. In addition, most of the nodes usually use batteries which may tempt a node to behave in a selfish manner when using its limited power supply.

2.3.4 Limited Physical Security

MANETs are generally more prone to physical security threats than other wireless networks because of their distributed system nature [3]. Routing algorithms for MANETs usually assume that nodes are cooperative and non-malicious [2]. As a result a malicious attacker can easily become an important routing agent and disrupt network operation. MANETs have an increased possibility of eavesdropping, spoofing, masquerading, and DoS type attacks.

2.4 MANET Routing Protocols

MANETs introduce more challenges in designing routing protocols than wired networks. Many protocols have been designed and developed to route data from a source to a destination under the limitations of these networks. A route is needed by a source whenever data needs to be transmitted to a destination through other intermediate nodes. MANET routing protocols are designed based on an absence of a centralized entity to create loop-free routes while keeping the communication overheads to a minimum as the topology changes [3]. Basically, routing protocols can be classified into [3]:

(a) Proactive Protocols

(b) Reactive Protocols

(c) Hybrid Protocols

(d) Multipath Protocols

2.4.1 Proactive Routing Protocols

Proactive MANET protocols are table-driven protocols that actively determine the layout of the network [3]. Each node maintains one or more tables containing routing information to other nodes in the network. A node frequently updates these tables to maintain the latest view of the network topology by propagating periodic updates to minimize route selection time. Mobility of nodes in MANET produces many

stale routes as the topology changes. So, these protocols are designed to optimize solutions that resolve the trade-off between maintaining an updated view of the network topology and a large amount of traffic overhead generated when processing these stale routes. Thus, proactive MANET protocols best suit small networks that have low node mobility. Some of the proactive routing protocols are presented in the following sections.

2.4.1.1 Destination Sequenced Distance-Vector (DSDV)

Destination-Sequenced Distance-Vector (DSDV) [7] is one of the earliest MANET routing protocols which is based on the Bellman Ford algorithm. A node maintains a routing table that includes a destination, a distance in hops to that destination and a sequence number which is assigned by the destination. Sequence numbers are used to determine stale and loop-free routes. A node periodically broadcasts its routing table to its neighbours using either a full or incremental dump. While a node sends its entire routing in full dump, it sends only those routes which have been changed since the last full dump as an incremental dump. Full dumps are preferable when there is little movement in the network and updates are infrequent. On the other hand, incremental dump is more efficient when the network is stable. Broken links can be detected by missing transmissions. A node discovers a broken link has to broadcast an update packet to inform others about this broken link.

2.4.1.2 Wireless Routing Protocol (WRP)

Wireless Routing Protocol (WRP) [8] is also based on the Bellman-Ford algorithm. A routing table includes a destination, a distance in hops to that destination and a cost metric to use this route which is usually the minimum cost of all the routes. To maintain a routing table, a node has to periodically send update packets to its neighbours that contain the recent route changes in its routing table. A node sends an empty HELLO packet if there is no route changes in its routing table to keep a link to a neighbour alive.

2.4.1.3 Fisheye State Routing (FSR)

Fisheye State Routing (FSR) [9] is designed based on the link state routing to reduce routing overhead in dynamic environments. Link state routing broadcasts the updated information throughout the network. A node has to update its routing table by the most accurate routing information about its closest neighbours only and exchange this information periodically with its neighbours. That ensures that a packet is routed along the most accurate route to a destination.

2.4.1.4 Optimized Link State Routing (OLSR)

Optimized Link State Routing (OLSR) [10] is designed based on the link state routing. This protocol suggests that a node elects a set of one-hop neighbours to act as multipoint relays (MPR). MPR has to be chosen such that its range covers all its two-hop neighbours. MPR forwards messages during the route information flooding process, generates link state information and reports links between themselves and their MPR electors. Non-MPR nodes process packets but do not forward those as the forward process is assigned only to MPRs. Bidirectional links can be determined by periodic HELLO packets containing information about all neighbours and their link status. A route is a sequence of hops from a source to a destination through MPRs within the network.

2.4.2 Reactive Routing Protocols

Reactive MANET protocols are on-demand protocols that are initiated by a source [11]. Routes are created whenever a source node requires to send data to a destination node. The source node initiates a route discovery procedure by transmitting route requests throughout the network and waits until receiving a reply from the destination node or an intermediate node that has a fresh route to that destination. An established route is maintained as long as it is required through a route maintenance procedure. The overheads added by these protocols include significant delay before the packet can be transmitted and a significant amount of control traffic.

Thus, reactive MANET protocols suit the networks that have high node mobility. Some reactive routing protocols are presented in the following sections.

2.4.2.1 Dynamic Source Routing (DSR)

Dynamic Source Routing (DSR) [11] is a source routing protocol which means that a data packet has to include a list of nodes representing the route to be followed. When a source node wants to send data to a destination, it firstly checks its route cache. If the required route is available, the source node includes the routing information inside the data packet before sending it. Otherwise, the source node initiates a route discovery by broadcasting a route request RREQ packet. Receiving a route request packet, a node checks its route cache. If the node does not have routing information for the requested destination, it appends its own address to the route before rebroadcasting the RREQ. The destination or an intermediate nodes generate a route reply RREP packet that includes the list of addresses received in RREQ and unicasts it back along this path to the source. Route maintenance in DSR can be achieved through sending RERR to the source node so it can initiate a new route discovery.

2.4.2.2 Ad hoc On Demand Distance Vector (AODV)

Ad hoc On Demand Distance Vector (AODV) [12] is designed to combine the features of both DSR and DSDV routing protocols. When a source node requires to send data to a destination and does not have a fresh route to this destination, it initiates a route discovery by broadcasting a route request RREQ packet. Each node that receives RREQ sets up a reverse route towards the source unless it has a fresher one. The destination or an intermediate node that has a fresh route to the destination unicasts a RREP packet which travels along the reverse path established at intermediate nodes during the route discovery process. Then the source node starts sending data to the destination through the neighbour node that first responded with a RREP. Route maintenance is accomplished by sending a route error RERR packet to the source if either the destination or one of the intermediate node moves away. Once the source node receives RERR, it re-initiates the route discovery process. AODV

uses a destination sequence number to ensure the freshness of a routing packet such as DSDV protocol.

2.4.2.3 Associativity-Based Routing (ABR)

Associativity Based Routing (ABR) [13] considers route stability as the most important factor in selecting a route. ABR maintains the routing table by using an associativity ticks mechanism. Periodic HELLO packets are exchanged between neighbour nodes. Every node keeps an associativity table, in which it records the connection stability between itself and its neighbours over time. A node that receives a HELLO increases the associativity tick of its source. Therefore, the link with a higher associativity tick is more stable than the one with a lower associativity tick. When a neighbour node moves out, the node resets its respective associativity tick. Routes are discovered by sending a broadcast query request packet which enables the destination to be aware of all possible routes between itself and the source. Once a destination has received the broadcast query packets, it selects the route with the highest degree of associativity.

2.4.3 Hybrid Routing Protocols

Hybrid routing protocols are designed to achieve advantages of both reactive and proactive routing protocols. These protocols use proactive protocols in areas that have low mobility nodes while they use reactive protocols in areas that have high mobility nodes. The hybrid protocols are proposed to reduce the control overhead of proactive routing approaches and decrease the latency caused by route search operations in reactive routing approaches. Some of the hybrid routing protocols are presented in the following sections.

2.4.3.1 Zone Routing Protocol (ZRP)

Zone Routing Protocol (ZRP) [14] consists of two routing protocols, Intra-zone Routing protocol (IARP) and Inter-zone Routing Protocol (IERP). Proactive protocol is used inside routing zones while reactive protocol is used between routing zones.

Neighbour discovery is accomplished by IARP to maintain up-to-date routing tables. IERP is used as a reactive protocol for connecting with nodes in different zones. ZRP is suitable for large networks to achieve a high performance by optimizing the transmission ranges of the nodes.

2.4.3.2 Core Extraction Distributed Ad hoc Routing Protocol (CEDAR)

Core Extraction Distributed Ad hoc Routing Protocol (CEDAR) [15] defines a subset of nodes in the network as "core". Routing messages are only broadcast among cores that can use any routing protocol. So, if a source node wishes to send data to a destination, it sends a route request packet to its local core asking for a route to the destination. The core is determined according to a distributed algorithm and the number of core nodes is kept small. To select core nodes, neighbouring nodes periodically exchange link state messages. Every mobile node in the network must be adjacent to at least one core node and picks this core node as its dominator. The algorithm guarantees that there is a core node at most 3 hops away from another core node.

2.4.4 Multipath Routing Protocols

Reactive routing protocols suffer a lot within large networks. Excessive routing overhead, high delay, unreliable data transfer and energy inefficiency are the main reasons for scalability problems of reactive routing protocols. Multipath routing is designed to improve the reliability by establishing multiple paths between a source and a destination [16]. These protocols generate disjoint paths which can be classified as node-disjoint paths and link-disjoint paths. While node-disjoint paths share only the source and the destination nodes in different links, link-disjoint paths do not share links but intermediate nodes might be shared. So, the route discovery mechanism of these protocols is designed to discover a maximum number of node-disjoint or link-disjoint paths [17]. After discovering all node-disjoint or link-disjoint paths, a node select a primary path from all these discovered paths. Some of the multipath protocols are presented in the following sections.

2.4.4.1 Ad hoc On-demand Multipath Distance Vector Routing (AOMDV)

Ad hoc On demand Multipath Distance Vector (AOMDV) [18], is an extension to the AODV routing protocol that is designed to create multiple loop-free and link-disjoint paths. AOMDV computes the multiple paths during the route discovery and it consists of two main components: a rule to create and maintain multiple paths at a node, and a distributed protocol to calculate the link-disjoint paths. Paths between a source and a destination are considered as disjoint paths if all intermediate nodes along the paths are different. Each route request and route reply packet received by a node has to be using a different route from the source to the destination. AOMDV uses one of the alternative paths to forward packets if a path to this destination is broken.

2.4.4.2 Scalable Multipath On-demand Routing (SMORT)

Scalable multipath on-demand routing protocol (SMORT) [19] is a multipath extension to AODV protocol. A primary path from a source to a destination is set up which is usually the shortest path and multiple paths are created as fail-safe paths. A fail-safe path is auxiliary to the primary path that bypasses at least one intermediate node on the primary path. Fail-safe path is used to send data when one or more of the bypassed nodes on the primary path leave the network. While a node can receive multiple copies of a Route Request (RREQ) packet, it rebroadcasts only the first copy of the RREQ if it has not a route to the destination. An intermediate node or a destination node can reply to multiple copies of RREQ. A node sending a RREP to the source has to select the best reverse path from the paths stored in the request table. Source node clarifies the first received RREP as the primary path and stores it in the routing table.

2.5 MANET Security

MANETs are more vulnerable to security attacks than fixed networks due their inherent characteristics. MANET mobility imposes a challenge that security services have to be provided regardless of a network topology change. Mobility adds as well a difficulty to distinguish between stale routes and fake routes and its dynamic topology changes add more opportunities for attacks. Security solutions of wired networks cannot be applied directly to MANETs because of the nature of the network and because of an absence of a fixed or central infrastructure which eliminates the possibility of establishing a centralized authority or trusted third party. Mobile nodes in MANETs have limited computation and limited battery power capabilities that impose another difficulty to implement cryptography and key management algorithms which need high computations such as public key algorithms. So, MANETs have security challenges that motivate researchers to resist different types of attacks from passive eavesdropping to active interfering. A malicious node can attack in different ways, such as replaying routing messages several times and fabricating fake routing information to disrupt routing operations.

MANET routing protocols are designed based on the assumption that all nodes cooperate without maliciously disrupting the operation of the protocol. However, the existence of malicious nodes cannot be ignored especially in MANETs because of the wireless nature of the network. Routing protocols in MANET can be disrupted by internal or external malicious nodes. An internal malicious node can be any legitimate node of the network while an external malicious node is any other node. However, external malicious nodes can be prohibited using cryptographic solutions such as digital signatures to authenticate the nodes. Internal malicious nodes have the capability to fully access the wireless link to propagate erroneous routing information, or to replay old routing information in order to partition a network or overload a network with retransmissions, thereby causing congestion, and hence a DoS. Detection of these malicious nodes through routing information is also difficult due to the dynamic topology of MANETs.

MANET routing protocols determine how data is forwarded from a source to a destination through a number of intermediate nodes. A routing protocol must encapsulate an essential set of security mechanisms that help prevent, detect, and respond to security attacks. Security requirements such as availability, confidentiality, authentication, integrity and non-repudiation [20] can be applied in designing routing protocols to limit these risks and to maintain a reliable and secure data transmission. MANET routing protocols cannot apply some of these goals due to the nature of the network.

Confidentiality: Confidentiality is sometimes called secrecy or privacy. It ensures a protection of information from being exposed to unauthorized access. The wireless nature of MANET introduces an opportunity to nodes within the direct transmission range to obtain the data. In addition, intermediate nodes which act as routers receive this data, so they can easily eavesdrop. MANET routing protocols do not include confidentiality as routing packets need to be processed by intermediate nodes before forwarding to other nodes in the network [21].

Availability: It ensures that a network service or resources are available to legitimate users when required in a timely manner and the network is survivable under malicious behaviours. For example, MANETs routing protocols can provide alternative routes to resist the DoS attack.

Authentication: It verifies the identity of a node to prevent impersonation. In wired networks and infrastructure-based wireless networks, it is possible to implement a central authority at a point such as a router, base station, or access point. On the other hand, MANETs cannot authenticate an entity because the lack of a central authority in these networks.

Integrity: It ensures that data has been not illegally altered during transmission. Due to the wireless nature of MANET, routing information such as hop count can be modified or deleted by a malicious node during transmission. A malicious node can also resend stale data which is known as a replay attack.

Non-repudiation: It ensures that a node cannot falsely deny having received or sent certain data. This is helpful when a node has proven its involvement in a malicious behaviour and its identity is sent to other nodes to ignore its upcoming data.

2.5.1 MANET Routing Attacks

Absence of a centralized authority and the assumption that nodes cooperate without maliciously disrupting the routing protocol make the process of securing routing protocols in MANET a challenging issue. MANET routing protocols do not have a clear line of defence to these attacks although these networks are accessible to both legitimate users and malicious attackers because of their wireless nature. Deliberate non-cooperation is mainly caused by either a selfish node that aims to save its power or a malicious node that is attempting to attack the network. Analysing various types of attacks that can be fabricated by either malicious or selfish nodes is always the key step towards developing good security solutions. MANET routing protocols are exposed to both active and passive attacks.

2.5.2 Passive Attacks

A Passive Attack does not disrupt the operation of a routing protocol, but tries to discover valuable information by eavesdropping data exchanged in a network without altering it. Therefore, a passive attack is a threat to data confidentiality. Detection of these attacks is impossible in most situations since the date being transmitted is not altered and hence defending against this type of attacks is complicated. Even if it is not possible to identify the exact location of a node, one may be able to discover information about the network topology, using these attacks. Some of these passive attacks are presented below.

2.5.2.1 Traffic Analysis

Traffic analysis is used to achieve information on which nodes communicate with each other and how much data is processed.

2.5.2.2 Traffic Monitoring

Traffic Monitoring is used to identify the two communicating nodes and functionality which could provide information to launch further attacks.

2.5.2.3 Eavesdropping

Eavesdropping usually happens in MANETs to obtain some confidential information that should be kept secret during the communication. This information may include the location, public key, private key or even passwords of the nodes.

2.5.3 Active Attacks

An Active attack occurs when a malicious node actively intercepts the data and attempts to modify, delete, add or redirect this data to disrupt the operation of the protocol. These attacks can be launched by external malicious nodes that do not belong to the network or by internal legitimate nodes that attempt to misbehave. A malicious node aims to cause congestion, propagate fake routing information or disturb nodes from providing services in external attacks. On the other hand, a malicious node aims to gain normal access of the network and shares network activities by impersonating legitimate node in internal attacks. Internal attacks are more severe and harder to detect than external attacks as they have legitimate access to the network. Routing protocols should be able to secure themselves against both types of attacks. Active attacks against MANET routing protocols are classified based on modification, impersonation or fabrication.

2.5.3.1 Modification-based Attacks

Modification Attacks [4] aim to violate data integrity by modifying routing packets to redirect them to specific nodes. Some of these attacks are presented below.

2.5.3.1.1 Redirection Attack

A destination always selects a route to a destination that depends on a metric value such as sequence number, hop count, delay etc. A malicious node in the redirection

attack [22] usually changes one or more of these metrics to spoof its neighbours that it has an optimum route to a destination in order to redirect the traffic.

2.5.3.1.2 Misrouting Attack

A malicious node in the misrouting attack [23] sends a data packet to the wrong destination. This type of attack is carried out by modifying the final destination address of the data packet or by forwarding a data packet to the wrong next hop in the route to the destination.

2.5.3.1.3 Detour Attack

A malicious node in a detour attack [24] adds a number of nodes to a route during the route discovery phase which diverts the traffic through these nodes that might include malicious nodes which could launch other attacks.

2.5.3.1.4 Blackmail Attack

Blackmail attack [22] causes false identification of a good node as a malicious node. MANET nodes usually keep information of discovered malicious nodes in a blacklist. A malicious node may blackmail a good node by telling other nodes in the network to add that node to their blacklists as well, thus avoiding this victim node in future routes.

2.5.3.1.5 Denial of Service (DoS) Attack

A malicious node in the DoS attack [25] aims a complete destruction of the routing function. A malicious node intercepts the route packet and modifies it before forwarding it to a next node which can cause the dropping of network traffic, redirecting to a different destination or to a longer route to reach to destination that causes unnecessary communication delay. Also a DoS attack can be implemented if a malicious node uses the routing protocol to advertise itself as having the shortest path to a destination which is a consequence to blackhole attack.

2.5.3.2 Impersonation-based Attacks

Impersonation attacks are a severe threat to the authenticity and confidentiality of MANET. Since current MANET routing protocols such as AODV and DSR do not authenticate source IP or MAC addresses, a malicious impersonates a legitimate node can launch many attacks by spoofing a routing packet. As a result, the malicious node can receive packets intended to the impersonated node or it can even completely isolate that impersonated node from the network. Some of these attacks are presented below.

2.5.3.2.1 Man-in-the-Middle Attack

A malicious node reads and possibly modifies the routing packets between two parties in this attack [26]. An attacker can impersonate the receiver with respect to the sender, and the sender with respect to the receiver, without having either of them realize that they have been attacked.

2.5.3.2.2 Sybil Attack

A malicious node in the Sybil attack [27] pretends to have multiple identities. An attacker can behave as if it were a larger number of nodes either by impersonating other nodes or simply by claiming false identities. All Sybil identities can participate simultaneously in the network or they may be cycled through.

2.5.3.3 Fabrication-based Attacks

Fabrication attacks are used to drain off the limited resources of nodes or the network connectivity by sending false routing packets. For example, flooding a specific node with unnecessary routing packets, fabricating route error RERRs to the other nodes so as to inform them that a route is no longer available, or attempting to create routes to nodes that do not exist which overwhelms the routing tables of neighbours, preventing them from creating of new routes. These attacks can result in a DoS attack. Some of these attacks are presented below.

2.5.3.3.1 Routing Table Poisoning Attack

A malicious node creates fictitious, or modifies genuine, routing packets before forwarding them to authorized nodes in the network to launch this attack [28]. This inflates the routing tables of victim nodes with a number of false entries and may result in congestion in portions of the network, or even make some parts of the network inaccessible.

2.5.3.3.2 Flooding Attack

A malicious node in a flooding attack [29] floods its neighbours with a high number of false requests RREQ to non-existent nodes. It aims to prevent its neighbours from creating new genuine routes by overwhelming their routing tables or by effectively using the bandwidth and processing resources of the nodes along the route.

2.5.3.3.3 Rushing Attack

A malicious node in a rushing attack [30] tampers the incoming RREQ packets by modifying the node list to include itself, and fast forwarding the modified RREQ packet to the next node. Since most of the reactive routing protocols forward only one RREQ packet, often the route request forwarded by the malicious nodes reaches the destination before all its copies from the other nodes which leads to inclusion of this malicious node in the created route between the source and the destination.

2.5.3.3.4 Blackhole Attack

A malicious node in a blackhole attack [31] advertises that it has an optimum route for a destination. This malicious node sends a modified packet or generates fake routing information to neighbour nodes causing other nodes to route data packets through this malicious one. Then, the malicious node discards any or all of the network traffic being routed through it. This attack can cause a DoS if the malicious node becomes dominant in multiple routes between sources and destinations.

2.5.3.3.5 Grayhole Attack

A malicious node in a grayhole attack [32] drops all data packets but it processes and forwards normally all control messages routed through it. So, the malicious node in reactive routing protocols replies with a genuine RREP clarifying that it has a route to a destination and later after asking by the source to route the data, it drops these data. This selective dropping makes grayhole attacks much more difficult to detect than blackhole attacks.

2.5.3.3.6 Wormhole Attack

A wormhole attack [33] involves the cooperation between two malicious nodes using a high-speed channel, such as a fast LAN connection, as a tunnel known as a wormhole. Wormhole attack is a severe threat to MANET routing protocols because it is hard to detect and the two colluding nodes can potentially distort the topology in which two distant nodes consider themselves neighbours and send data using the tunnel [34].

2.5.3.3.7 Selfish Attack

A malicious node in a selfish attack [1] saves its resources, such as battery, by not cooperating in the network operations. The selfish node drops all data and control packets even if these packets are sent to it which affects the network performance. It complies with routing protocol when it requires to send data to a destination.

2.6 Securing MANET Routing Protocols

MANETs secured routing protocols mainly use various cryptographic algorithms to achieve secured routing information. Before we discuss some of the available secured routing mechanisms, we briefly present the cryptographic algorithms in the following section.

2.6.1 Cryptographic Algorithms

A cryptographic algorithm is a sequence of processes used to encipher and decipher packets [35]. Encryption and decryption allows communications between two parties while preventing unauthorized third parties from understanding their communications. Often cryptographic algorithms are necessary to secure networks, particularly when communicating through an untrusted network such as the Internet or an open medium network such as a MANET. A source encrypts plaintext into ciphertext before sending this ciphertext through the network, while a destination decrypts this ciphertext to restore the plaintext. Cryptographic algorithms can be classified as symmetric key encryption, asymmetric key encryption, cryptographic hash functions and digital signatures.

2.6.1.1 Symmetric Key Cryptography

Symmetric key algorithms [36] encrypt and decrypt data using a single key, which is normally called a secret key, as shown in Figure 2.2 [37]. This shared key must remain secret to be effective. Sending a secret key as a plain text through a medium before using it increases the possibility of an attacker to record this key for later use. Secure key management and distribution is a challenge in large networks as it involves a large number of keys' authentications and secure distribution of them. Exchanging secret keys is difficult, especially in wireless networks, since each two communication parties usually exchange keys on the same medium that they are using encryption to protect from. Cryptographic key exchange protocols such as the Diffie-Hellman protocol introduce a solution for this key distribution problem by allowing key agreement without revealing the key on the network [35]. Triple Data Encryption Standard (3DES) [38], Advanced Encryption Standard (AES) [39], RC4 and RC5 [40] are the most commonly used symmetric key algorithms. To secure data using symmetric key encryption, the current advice is that the key length should not be less than 90 bits [41].

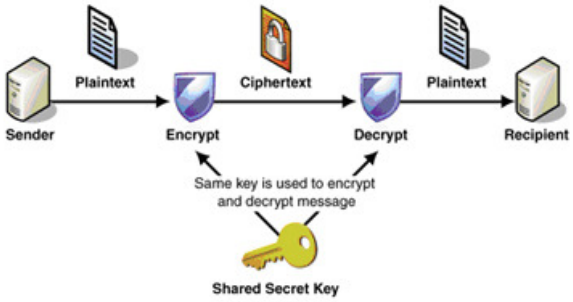

Figure 2.2: Symmetric Key Cryptography Model [37]

2.6.1.2 Asymmetric Key Cryptography

Asymmetric key algorithms, known also as public key algorithms, use two keys as shown in Figure 2.3 [37]. These keys are mathematically related although knowledge of one key does not introduce an opportunity to easily determine the other key [35]. A node has two keys, the public key is the key that can be freely distributed to other nodes and used by these nodes to encrypt data for this node. The other key is the private key that is only known by this node and is used to decrypt the data for itself. Public keys are usually delivered to nodes with certificates that are validated by trusted third parties. Public key cryptography is rarely used in direct data encryption because it is slow for large messages [1]. Generally, public key encryption is used to agree on a secret key for a symmetric algorithm, and then all further encryption is done using this secret key. Therefore, public key encryption algorithms are primarily used in key exchange protocols and when non-repudiation is required. Rivest-Shamir-Adleman (RSA) [42] and Elliptic Curve Cryptography (ECC) [43] are the most popular public key encryption algorithms. RSA, Diffie-Hellman, and El Gamal techniques require more bits for the keys for equivalent security compared to typical symmetric keys; a 1024-bit key in these systems is supposed to be roughly equivalent to an 80-bit symmetric key [41]. Today, RSA users generally use keys that are at least 2048 bits long [44].

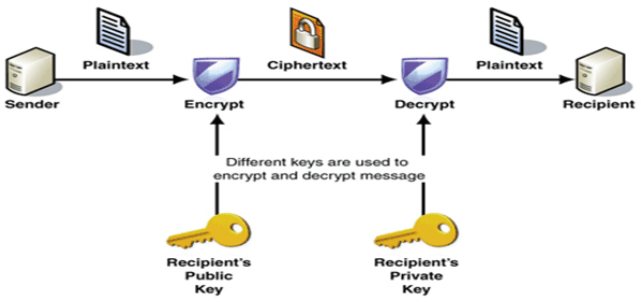

Figure 2.3: Asymmetric Key Cryptography Model [37]

2.6.1.3 Cryptographic hash functions

Cryptographic hash functions are essentially cryptographic checksum algorithms with special properties that produce a message digest known as Message Authentication Code (MAC) [45]. It is an algorithm which creates a standard length digital representation of a message which is usually much smaller than the message and unique to it. A change to the message will produce a different hash result even when the same hash function is used. A cryptographic hash is a one-way function meaning that given the hash value it is computationally infeasible to reconstruct the original message. Usually, a message digest of a message is generated by a sender as a function of a secret key and a hash function and then it is appended to the message. Once receiving a message, a recipient regenerates the message digest of the received message and compares it to the appended digest in the message to ensure the integrity of the message. Message Digest (MD5) [46] and Secure Hash Algorithm (SHA) [47] are the most popular one-way cryptographic hash functions.

2.6.1.4 Digital signatures

MAC does not suit many applications because they require agreeing on a shared secret. A digital signature introduces a solution that depends on public key cryptography. A source node signs a message with its private key, then anyone can use the source's public key to verify that the source is who has signed the message. The RSA digital signature [42] and the Digital Signature Algorithm (DSA) [48] are the most popular digital signature algorithms. Digital signature algorithms are very slow as they are designed based on public key algorithms. To overcome this draw-

back, the algorithm generally is applied to a message digest instead of the entire message. An application using digital signatures is the public key certificate created by trusted third party.

2.6.2 Secured Routing Mechanisms

The majority of MANET routing protocols have been designed without security considerations. As mentioned earlier, their design assumes that nodes are trusted and they cooperate without maliciously disrupting the protocol operation. However, existence of malicious nodes cannot be ignored in a large scale and dynamic network such as a MANET. Non-securing MANET routing protocol exposes nodes to several different types of attacks. MANET secured routing protocols can be mainly categorized in two major categories:

Prevention: Prevention mechanism is designed to prevent a malicious node from initiating a misbehaviour action. Its design is based on modifying a routing protocol by including the cryptographic algorithms. It usually appends encrypted routing packet information as an extension to the basic routing protocol's packet to authenticate the confidentiality, integrity and non-repudiation of routing information. These proposals usually use symmetric, asymmetric or hash algorithms.

Detection and Reaction: Detection and Reaction mechanisms attempt to discover malicious nodes or misbehaviour actions in the network to maintain the proper routing protocol operation. In addition to various types of attacks, there are a large number of routing protocol misbehaviours that have to be discovered. A selfish node may misbehave by agreeing to forward packets and then refrain from doing so to save its resources. A malicious node launches a DoS attack by dropping packets. So, Detection and Reaction mechanisms have to detect and react to such misbehaviours.

2.6.2.1 Prevention Mechanisms

These mechanisms use cryptographic algorithms to prevent various attacks. One of the challenges for these solutions, if using an asymmetric key or a digital signa-

ture, is the management and distribution of the keys as a MANET does not have a centralized trusted third party. Another challenge for these solutions, if using symmetric key, is the large number of secret keys that have to be exchanged between nodes. Mathematically, a network of **n** nodes requires **n(n+1)/2** pairs of secret keys in the network. Another challenge is that these keys are usually transmitted through the wireless medium which introduces an opportunity for a malicious node to intercept one and use it later to launch an attack. Hash algorithms are used to ensure the integrity of the message. One of the most important disadvantages of the prevention mechanisms is that they cannot ensure complete cooperation among nodes in the network.

2.6.2.1.1 Authenticated Routing for Ad-hoc Networks (ARAN)

Authenticated Routing for Ad-hoc Networks (ARAN) [49] is a secured reactive routing protocol that implements cryptographic certificates to achieve end-to-end authenticated routing information. It assumes pre-establishment of key management and distribution and requires that a node has to have a preliminary certification signed by a trusted certification authority before joining the network.

A source initiates a route discovery by broadcasting a Route Discovery Packet (RDP) that is digitally signed by its private key. RDP includes the certificate of the source node, a nonce, a timestamp and the address of the destination node. A nonce and timestamp are used to prevent replay attacks and to ensure loop-free routes. A neighbour of the source has to verify the integrity of the received packet by verifying the signature of the source before appending its own signature encapsulated over the signed packet. All subsequent intermediate nodes remove the signature of their predecessor, verify it before appending their signature to the packet. The destination verifies the signature of both its predecessor and the source before sending a Reply packet (REP). Similarly, each node along the path from the destination to the source has to verify the signature of its predecessor, sign and append its own certificate before forwarding the REP to the next hop. A source verifies the signature of both its predecessor and the destination received in the REP and trust the returned path

from destination. Route maintenance can be achieved via an error message that is generated and forwarded to the source node if there is a broken link.

ARAN provides effective protection from modification, impersonation, and fabrication attacks as a result of its strong authentication, message integrity, and non-repudiation features. On the other hand, ARAN has a high computational cost for using heavy asymmetric cryptographic operations and large routing packets. Selfish attacks cannot be discovered by ARAN.

2.6.2.1.2 Security-Aware ad-hoc Routing (SAR)

Security-Aware ad-hoc Routing (SAR) [50] is an extension of the AODV routing protocol that implements symmetric cryptography to provide security to a MANET routing protocol. Different trust levels are implemented using shared symmetric keys between nodes along a path. Nodes that have a same trust level should have the same shared secret key. A node has to encrypt a routing packet using a shared secret key before sending it. Once receiving this routing packet, a node has to decrypt it using the shared secret key before rebroadcasting it. So, only nodes with the same trust level can read and forward a packet. A source has to select the desired security level to route the data. A packet received by a destination has to travel through nodes having the same trust level as the source node. A RREP is sent from an intermediate node or the destination node to the source node through the same shared secret key. If there is more than one route sharing the same secret key, the shortest route is selected for data forwarding. SAR provides effective protection of routing messages from modification, impersonation, and fabrication attacks. The disadvantage of SAR is that it involves significant encryption overhead since source, intermediate and destination nodes have to perform both encryption and decryption operations.

2.6.2.1.3 Secure Routing Protocol (SRP)

Secure Routing Protocol (SRP) [51] is a secured reactive routing protocol designed based on the assumption that each pair of nodes share a secret key. The protocol uses symmetric and hash algorithms to authenticate the IP header, routing information

and the shared key. A source node sends a RREQ packet that includes the hashed value of the request, which is calculated based on the agreed secret key between them, to a destination. The destination generates the hashed value of the request and compares the output with the one received from the source. Matching ensures the authenticity of the sender and the integrity of the RREQ message. The destination replies with a RREP packet that includes the path information from the source to destination and appends the hash value of the reply using their agreed secret key. The source node, upon receiving the RREP packet, generates the hash value of the reply and compares it to the received one to ensure the identity of the destination and the integrity of the reply. If they match, the source starts sending the data encrypted using their secret key. SRP achieves a low computational cost and overhead and successfully protects against IP spoofing because the IP address is one of the values hashed.

2.6.2.1.4 Secure Efficient Ad hoc Networks (SEAD)

Secure Efficient Ad hoc Networks (SEAD) [52] is a secured proactive routing protocol designed based on the DSDV protocol. The protocol implements symmetric and hash algorithms to protect the modification of routing information such as metric, sequence number and source node. SEAD uses a hash for ensuring the integrity of the information encapsulated in the routing updates. Authentication of the source is achieved by providing a signature of the hash value using a pre-established shared secret key between the nodes to authenticate the routing update message. Creating loop-free routes can be achieved by using destination sequence numbers to protect against reply attacks. The protocol provides strong protection against malicious nodes attempting to tamper the destination sequence number. However, SEAD does not protect against an attacker tampering the next hop or the destination field of a routing update packet as they are not hashed.

2.6.2.1.5 ARIADNE

ARIADNE [53] is a secure reactive routing protocol that implements symmetric and hash algorithms and its design is based on DSR to provide authentication of

routing messages. ARIADNE assumes that an agreed shared secret between each pair of nodes is established before starting its normal operation. A source node sends a RREQ containing its address, a destination address, timestamp, and hashed value for this information. An intermediate node verifies the hash value before appending its address to the node list and replaces the old hash value with a new one that includes its address. The destination node verifies each hop of the route by comparing the received hash value with the generated hash value of the DSR packet and then replies with RREP. The source repeats the process to authenticate the reply from the destination. ARIADNE provides good defence against modification, fabrication, and spoofing as a result of routing message verification feature.

2.6.2.1.6 Secure Ad hoc On-demand Distance Vector Routing Protocol (SAODV)

Secure Ad-hoc On-demand Distance Vector Routing Protocol (SAODV) [54] is a secured reactive routing protocol that implements asymmetric and hash algorithms and its design is based on AODV. SAODV provides an end-to-end authentication and hop-to-hop verification of the routing messages. SAODV uses digital signature to authenticate a routing message and a hash function to authenticate the hop count field within RREQ and RREP messages as it is the only mutable field.

A source sending a RREQ creates a hash value for the hop count and signs the request with its private key. An intermediate node verifies both the hop count and the digital signature of the RREQ by the public key of the source. Successful verification ensures the identity and integrity of the request and the node then can establish the reverse route entry to the source in its routing table. Then the intermediate node increments the hop count value of the RREQ and computes the new hash value that includes the new hop count before appending it to the RREQ and broadcast. The destination verifies both the hop count and the digital signature of the RREQ to ensure the identity and integrity of the request and replies with a RREP that is signed with its private key. Similarly, intermediate nodes and the

source verify the RREP similarly to authenticate the identity and the hop count of the destination.

2.6.2.1.7 Secure Link State Routing Protocol (SLSP)

Secure Link State Routing Protocol (SLSP) [55] is a secure proactive protocol that implements asymmetric and hash algorithms to secure the route discovery and the distribution of link state information. Periodic Hello messages are used in the neighbour discovery phase and are signed by the private key of the sender. A node receiving a link update messages verifies the attached signature using the public key of the sender. The hop count field in the update message is protected using a hash function.

2.6.2.2 Detection and Reaction Mechanisms

Detection and Reaction mechanisms discover malicious nodes and take a proper action to maintain the proper operation of the routing protocol. Byzantine algorithm, CORE, CONFIDANT, and Watchdog and Pathrater are examples of these algorithms.

2.6.2.2.1 Byzantine Algorithm

The Byzantine algorithm [56] is used to protect the network from byzantine failures which include the modification of packets, dropping packets and attacks caused by selfish or malicious nodes. The Byzantine algorithm consists of three different phases which are route discovery, fault detection and link weight management. The route discovery phase is very similar to reactive protocols in which a source node broadcasts a RREQ packet containing its address, destination address, a sequence number, and also includes a list of detected malicious links and their weights signed with its private key for authentication to its neighbours. An intermediate node that does not have a route to the destination verifies the signature of the source before rebroadcasting it to other nodes. The destination node verifies the signature and creates a route reply message RREP towards the source that contains the source,

the destination, a respond sequence number and a combined link weight list (for both the source and the destination).

The fault detection phase ensures that an intermediate node sends an acknowledgement to the source node for every received packet. If the number of unacknowledged packets exceeds some threshold value, the route is registered as faulty. The link weight management phase calculates the weight of the links. If a link is identified as faulty by the fault detection phase, its corresponding link weight value is doubled which affects on the following route discovery phases as a link with a lower weight value is considered as better link in the route discovery phase .

2.6.2.2.2 CORE

CORE [25] is a protocol that monitors the cooperative behaviour of nodes. It uses a reputation table and Watchdog mechanism to identify the misbehaving nodes. The reputation table maintains a table of intermediate nodes and their associated reputations. The Watchdog calculates the reputation value based on a reputation function. Whenever an intermediate node refuses to cooperate with a source node, the CORE scheme will decrease the reputation of this intermediate node which can lead to the elimination of this intermediate node from the network when its reputation falls below a threshold.

2.6.2.2.3 CONFIDANT

The Cooperation of Nodes Fairness In Dynamic Ad hoc Networks (CONFIDANT) [57] protocol is used to identify non-cooperative nodes. This protocol consists of a monitor, a trust manager, a reputation system and a path manager. The monitor is responsible for monitoring passive acknowledgements for each packet it forwards. The trust manager is responsible for sending and receiving alarm messages. Alarm messages are exchanged between nodes that are predefined as friends about misbehaving nodes.

The reputation system maintains a table of nodes and their associated ratings. Ratings are modified according to a rate function that assigns a substantially smaller weight to alarm messages reported from friends regarding a misbehaving node and

a greater weight for direct observations. The path manager manages path addition, deletion, and update according to the feedback it received from the reputation system. If a rating falls under a certain threshold the path manager removes the path containing an identified malicious node.

2.6.2.2.4 Watchdog and Pathrater

The Watchdog and Pathrater [58] protocol is used to discover malicious nodes by monitoring the behaviour of next node in a path. A node transmitting a packet to its next node in a path eavesdrops on the medium to ensure that this next node retransmits the packet without altering it. Watchdog increases the failure rating of a node if it notices a malicious behaviour such as DoS or modification of the packet. This failure rating is used by Pathrater to determine a reliable path from a source to a destination.

2.6.2.3 Secured Routing Mechanisms Drawbacks

Prevention mechanisms are designed based on the usage of cryptographic algorithms which do not suit the limited resources of MANET nodes. In addition, these solutions succeed in discovering some attacks and fail in others especially the flooding attacks such as SAODV as shown later in Chapter 4. On the other hand, detection and reaction mechanisms are used to discover malicious nodes or misbehaviour actions in the network to maintain the proper routing protocol operation. These solutions cannot guarantee the true classification of nodes because the cooperative nature of the MANETs which leads to false exclusion of innocent nodes and/or good classification of malicious nodes as shown later in Chapter 5 and Chapter 6.

The drawbacks of both types of mechanisms are the key to study and investigate the impacts of different types of attacks on the MANET reactive routing protocols. The impact of flooding, blackhole, grayhole and selfish attacks on the MANET reactive routing protocol are presented in Chapter 4. As it will be clear, the blackhole and flooding attacks have dramatic negative impact while grayhole and selfish attacks have little negative impact on the performance of MANET routing protocols. This study and analysis clarifies that the current solutions for both

flooding and blackhole attacks as shown in Chapter 5 and Chapter 6 respectively have many drawbacks that prevent them from being implemented. These drawbacks introduce the motivation to design and implement new mechanisms that can resist both attacks.

2.7 Summary

MANETs introduce more challenges in designing routing protocols than wired networks. MANET routing protocols can be classified into proactive, reactive, hybrid and multipath routing protocols. Proactive MANET protocols are designed based on a node maintaining one or more tables containing routing information to other nodes in the network and has to frequently update these tables to maintain a latest view of the network topology. Reactive MANET protocols are initiated by a source and the routes are created whenever a source node requires to send data to a destination node. Hybrid routing protocols use proactive protocols in areas that have low node mobility while use reactive protocols in areas that have high node mobility. Multipath routing is designed to improve the reliability by establishing multiple paths between a source and destination.

MANET routing protocols has not a clear line of defence to either active or passive attacks as they are designed based on the assumption that nodes cooperate without malicious disruption of the routing protocol. Security solutions of wired networks cannot be applied directly to MANETs because of an absence of a centralized authority, and limited resources that impose another difficulty to implement cryptography and key management algorithms. MANETs secured routing protocols can be mainly categorized into prevention mechanisms, and detection and reaction mechanisms. Prevention mechanism use cryptographic algorithms to prevent a malicious node to initiate a misbehaviour action. Detection and Reaction mechanism attempts to discover malicious nodes or misbehaviour actions in the network to maintain the proper routing protocol operation.

Chapter 3

Methodology

3.1 Introduction

MANET routing protocols are designed based on the assumption that all nodes cooperate without maliciously disrupting the operation of the routing protocol. Reactive MANET protocols are initiated by a source whenever it needs to send data to a destination node. Selecting well-known reactive routing protocols that represent different approaches such as single path, multipath and secured from the wide span of these protocols requires a detailed study of these protocols. AODV, DSR, AOMDV and SAODV routing protocols are selected to study their behaviours under various attacks.

MANET routing protocols should be able to secure themselves against various attacks. Active attacks are classified based on modification, impersonation or fabrication of the routing packets. Selecting multiple attacks that represent fabrications or modifications of routing packets, partial or full misbehaving requires as well a deep study of various attacks. Flooding, blackhole, grayhole and selfish attacks are selected to represent the wide range of malicious behaviours and analyse the effect of them on network performance. NS-2 is the most widely used network simulation tool for network research. It is classified as an object-orientated discrete event simulator. NS-2 has been selected as a simulation tool to study the performance of different protocols in the presence of various attacks.

The rest of the chapter is organized as follows. Section 3.2 presents the details about the chosen MANET routing protocols. In Section 3.3, selected MANET routing attacks are presented. Section 3.4 discusses the simulation environment and modifications to the simulator. In Section 3.5, a summary is presented.

3.2 Routing Protocols

As we mentioned in Chapter 2, routing protocols can be classified mainly into proactive and reactive protocols. Proactive routing protocols are designed to maintain a routing table that includes routing information to all other nodes in the network. On the other hand, Reactive routing protocols are designed that routes are created whenever a source requires to send data to a destination. While proactive protocols suit small networks that have low node mobility, they suffer a lot in high dynamic networks. Reactive MANET protocols are designed to suit networks that have high node mobility which makes it a fertile area for research. So, our focus in the following chapters is directed to reactive routing protocols to analyse their behaviours.

3.2.1 Selection Criteria

Reactive MANET protocols are initiated by a source. Routes are created whenever a source node requires to send data to a destination node. The source node initiates a route discovery procedure by transmitting route requests throughout the network. The destination node or an intermediate node that has a route to the destination has to respond with a list of intermediate nodes between the source and the destination. An established route is maintained as long as it is required through a route maintenance procedure.

Our choice of AODV and DSR is based on their simplicity and popularity as reactive routing protocols for MANET. On the other hand, while AOMDV represents multipath routing protocols, SAODV represents secured routing protocols in MANET. AODV [12] is one of the extensively studied reactive protocols which were jointly developed on July 2003 in Nokia Research Centre, University of California, Santa Barbara and University of Cincinnati. DSR [11] is one of the earliest

and most well-known MANET reactive protocols. AOMDV is an extension to the AODV routing protocol designed to provide efficient recovery from route failures and efficient fault tolerance using multipath. SAODV [54] is an enhancement over AODV routing protocol that utilizes security features.

3.2.2 Dynamic Source Routing (DSR)

DSR [11] is one of the most well-known MANET reactive protocols. The protocol is an on-demand source routing protocol which implies that the data packets contain a list of nodes representing the route to be followed and the routes are created whenever a source node requires to send data to a destination node. DSR supports caching multiple routes to a single destination which enables a node to use any of these routes for data forwarding.

The protocol consists of two mechanisms which are route discovery and route maintenance. A source node wishing to send data has to consult its route cache for an available route to a destination. If a route to the destination is available, it includes the routing path inside the data packet before sending it. Otherwise, the source node initiates a route discovery operation by broadcasting a route request RREQ packet. An intermediate node that receives a RREQ checks its route cache for an available route to the destination sending a route reply RREP packet to the source if the route is available. If the node does not have a route for the requested destination, it appends its own address to the RREQ packet before rebroadcasting the RREQ packet to its neighbours. The destination node generates a RREP packet that includes the list of addresses received in the RREQ, unicasts it back along this path to the source and stores this route in its route cache for later use.

Route maintenance in DSR can be achieved through the confirmations that nodes generate when they can verify that the next node successfully received a packet. These confirmations can be link-layer acknowledgements, passive acknowledgements or network-layer acknowledgements specified by the DSR protocol. A node on a route is responsible for confirming that a packet has been received by a next node in the route and retransmitting the packet if necessary. If no confirmation is received

after a limited number of retransmission attempts for the packet, the link from this node to the next hop is considered to be broken, and the route maintenance mechanism sends a route error RERR packet to the source node identifying this broken link. The source node removes all the routes that include the failing link from its cache and uses an alternate route that it may already have or may re-invoke route discovery to discover a new route to this destination.

3.2.3 Ad hoc On Demand Distance Vector (AODV)

AODV [12] is a reactive routing protocol. It uses destination sequence numbers to ensure the freshness of routes and to guarantee loop-free routes. A node that receives a routing packet updates its routing table, with the routing information included in this routing packet, if it has a destination sequence number smaller than the one received (i.e. old route) or it has the same sequence number with a higher hop count (i.e. longer route). Non-active routes expire after a timeout and have to be removed from the routing table of a node. Routing information for a route is stored only in the source, the destination, and intermediate nodes along this active route which decreases the memory overhead and minimizes the use of network resources.

A source node requires to send data to a destination node has to first check its routing table for an available route to this destination. If it does not have a valid route, it initiates a route discovery process by broadcasting a route request (RREQ) packet to its neighbours using a new sequence number. An intermediate node that receives a RREQ discards it if it is either the source of the request or it has already rebroadcast the same RREQ earlier. Then, the intermediate node sets up a reverse route towards the source and updates its routing table unless it has a fresher one (i.e. higher destination sequence number) and increments the hop count before rebroadcasting the RREQ packet to its neighbours if it has not a fresh route to this destination. During the process of forwarding the RREQ, an intermediate node records in its routing table the address of the neighbour from which the first copy of the broadcast packet is received as a next hop towards the source, thereby establishing a reverse path.

The destination or an intermediate node with a route to the destination in its routing table unicasts a RREP packet back to the neighbour from which it received the first RREQ, which relays the RREP backward via the reverse path to the source node. An intermediate node sets up a forward route to the destination. This procedure enables all intermediate nodes of a discovered path to have routes to both the source and the destination. When the source node receives the RREP packet, it starts sending data packets to the destination node through the neighbour node that first responded with the RREP.

Routes maintenance can be achieved by HELLO messages that are broadcast periodically to neighbour nodes and a route error (RERR) packet that reports a link failure. When a source node moves, it has to reinitiate the route discovery protocol to find a new route to the destination if the next hop towards the destination becomes unreachable. On the other hand, when an intermediate node along the route moves, its upstream neighbour will notice route breakage due to the movement and broadcast a RERR to only its neighbours. An active neighbour in the route rebroadcasts the RERR packet to their upstream neighbours until the RERR is received by the source node. The source node may then choose to reinitiate the route discovery for that destination if a route is still desired.

3.2.4 Ad hoc On-demand Multipath Distance Vector Routing (AOMDV)

AOMDV [18], is an extension to AODV routing protocol that is designed to create multiple loop-free and link-disjoint paths. AOMDV computes the multiple paths during the route discovery phase that consists of a rule to create and maintain multiple paths and a distributed protocol to calculate the link-disjoint paths. Paths between a source and a destination are considered as disjoint paths if all intermediate nodes along the paths are different. AOMDV uses one of the alternative paths to forward packets if a path to this destination is broken.

A route request and a route reply packet received by a node has to be using different routes from the source to the destination. Using duplicate route request

RREQ copies, AOMDV creates loop-free reverse paths to the source at both intermediate and destination nodes. The destination generates multiple route replies RREP which have to be sent along multiple loop-free reverse paths that have been already created during the route request phase to create multiple loop-free forward paths to the destination. To select the best route to the destination and guarantee loop-free paths, *advertised_hop_count* is used which is the maximum acceptable hop count for any path at a node. Only paths with a hop count less than the advertised value is accepted while the remaining paths are discarded.

As AOMDV is an extension to AODV, it consequently shares some of its features such as using destination sequence numbers to ensure loop-free paths. In terms of route maintenance, the only difference between AODV and AOMDV is that while a node sends a route error RERR to the source when a link is broken in AODV, it sends this error if it has not an alternative path to the destination in AOMDV.

3.2.5 Secure Ad hoc On-demand Distance Vector Routing Protocol (SAODV)

This [54] is an enhancement of AODV routing protocol to fulfil security feature such as integrity and authentication. SAODV provides an end-to-end authentication and hop-to-hop verification of the routing messages. The protocol assumes that a node has certified public keys of other nodes in the network and a certified private key for itself. SAODV uses asymmetric cryptography to authenticate all non-mutable fields of routing messages and a hash algorithm to authenticate the hop count (the only mutable) field. The protocol suggests appending an extension message that includes a hash value of the hop count and a digital signature of the packet using the private key of the sender. A node can verify a signature of a sender using the sender's public key to ensure the identity of the sender and verify hash value of the hop count using the hash function that is included in the packet to ensure the integrity of the packet. A node fails to verify hash value or digital signature discards the received routing packet.

As the protocol design is based on AODV, it uses the same route discovery and route maintenance procedure. A source node sends a RREQ to a destination includes a hashed value of the hop count and a signature of the RREQ by its private key. An intermediate node that receives a RREQ has to verify the hash value of hop count and the digital signature. If it succeeded in verifying both of them, it stores a reverse route entry to the source in its routing table, increments the hop count value in RREQ packet, generates a new hash value and rebroadcasts the RREQ again to its neighbour. The destination that succeeded in verifications has to reply by sending RREP that includes a hashed value of the hop count and a signature of the RREP by its private key. Similarly, the source and intermediate nodes have to verify both the hash of the hop count and the signature of the RREP before adding the forward route to their routing tables. This procedure ensures that both the source and the destination can identify its communication partner and avoid impersonation attacks.

The above scenario implies that it is impossible for intermediate nodes to reply to RREQs even if they have a route to the destination because the RREP message must be signed by the destination's private key which is known only to the destination. To imitate AODV that permits to other nodes that have a fresh route to the destination to send a RREP, SAODV suggests a delegation feature that allows intermediate nodes to reply to RREQ messages. This delegation is based on a double signature in which a node sends a RREQ message can include a second signature that is computed on a fictitious RREP message towards itself. Intermediate nodes stores this second signature in their routing table to be used if later a node asks for a route to the owner of the double signature. Then, the intermediate node generates the RREP message includes the double signature and signs this message with its own private key.

Routes maintenance can be achieved by RERR messages which report a link failure. SAODV suggests that RERR messages have to be fully signed using the private key of its sender as these messages have a large amount of mutable information. A node that receives this RERR has to verify the signature to ensure the identity of

the sender. Since the destination sequence numbers in RERR are not signed by the corresponding node, a node should never update any destination sequence number of its routing table based on a RERR message.

Although SAODV does not require additional routing messages, SAODV messages are significantly bigger, mostly because of the digital signatures. Moreover, SAODV requires heavyweight asymmetric cryptographic calculations as result of generation and verification of the digital signature at each node which gets worse when the double signature mechanism is used as it implies the generation or verification of two signatures for a single message.

3.3 Routing Attacks

MANET routing protocols are exposed to active attacks and passive attacks. Passive attacks do not disrupt the operation of the routing protocol, but try to discover valuable information by eavesdropping data exchanged in network without altering it. On the other hand, active attacks occur when a malicious node actively intercepts the data and attempts to modify, delete, add or redirect this data to disrupt the operation of the protocol.

3.3.1 Selection Criteria

Routing protocols should be able to secure themselves against active attacks. Active attacks are classified based on modification, impersonation or fabrication to the routing packets. Our choice of flooding attack to represent the fabrication attacks in which a malicious node initiates the attack by generating a fake routing message. The malicious node constructs RREQ packets to overwhelm the network to launch a denial of service. A malicious node may also impersonate another node to convince its neighbours that it forwards a RREQ that is received from another node. Blackhole attack represents fabrication attacks in which a malicious node launches the attack by responding to a genuine routing message by a fake RREP packet that will lead to the direction of the traffic towards a node which will later drops all these data causing denial of service. Thus, the flooding attack represents a malicious node

action while the blackhole attack represents its reaction. Selfish attack represents a full misbehaviour of a malicious node and the loss of cooperation on which MANET design is based, in which the malicious node drops all the data and routing packets routed through it. Grayhole attack represents partial misbehaviour of a malicious node in complying with the protocol operation in which the malicious node drops the data packets while it follows the normal protocol operation for the routing packets.

3.3.2 Flooding Attack

In a flooding attack [29], a malicious node floods the network with a large number of RREQs to non-existent destinations in the network. Since these destinations do not exist in the network, a RREP packet cannot be generated by a node in the network. When a large number of fake RREQ packets are broadcast into the network, new routes can no longer be added and the network is unable to transmit data packets. This leads to congestion in the network and overflow of the routing table of nodes so that the nodes cannot receive new RREQ packet, resulting in a DoS attack. Moreover, unnecessary forwarding of these fake RREQ packets has serious effects in MANET [59] as a result of limited computational and power resources of nodes.

3.3.3 Blackhole Attack

In a blackhole attack [60], a malicious node absorbs the network traffic and drops all packets. Once a malicious node receives a RREQ packet from a node, it does not query its routing table about a route to the destination. Rather, it immediately sends a false RREP with a high sequence number and a minimum hop count to spoof its neighbours that it has the best route to the destination. Thus, the reply from the malicious node will be received by the source node before all other replies and a route that includes this malicious node is selected to send the data packets. Later, when data packets are routed through the blackhole node, it drops the packets rather than forwarding them to the destination node. This attack can cause DoS if the malicious node becomes dominant in the majority of routes between sources and destinations.

3.3.4 Grayhole Attack

In a grayhole attack [21], a malicious node behaves normally as a truthful node by replying with true RREP packets to the nodes that sent RREQ packets. It queries its routing table about an available route to the destination and replies with a RREP if available which fully complying with the protocol operation. If later data is routed through this malicious node, it starts dropping these data packets to launch a (DoS) denial of service attack. So, the malicious node forwards routing packets and drops data packets, which makes grayhole attacks much more difficult to detect.

3.3.5 Selfish Attack

In a selfish attack [1], a malicious node saves its resources; such as battery, by not cooperating in the network operations. A selfish node affects the network performance as it does not correctly process routing or data packets based on the routing protocol. The selfish node drops all data and control packets even if these packets are sent to it to save its resources. When a selfish node needs to send data to a destination, it starts following the normal routing protocol operation. After it finishes sending its data, the node returns to its silent mode and the selfish behaviour.

3.4 System Modelling

Implementing a real experiment on a testbed is costly, expensive, and time consuming. Therefore, using the system modelling is essential to better understand the behaviour of the protocol and evaluate the new idea before it can be implemented in the testbed and finally implemented in a real life experiment [61]. Moreover, most of the proposed algorithms and protocols in MANETs are designed to support hundreds of nodes which introduces the difficulty of implementing and testing their performances on testbed without examining their feasibility and results using modelling techniques.

System modelling uses a simple representation of an actual system to achieve predictions about how a system will behave without implementing it. Various param-

eters and often some simplification assumptions can be applied to study a system's behaviour. Modelling can be classified into two approaches; analytical approach and simulation approach [62]. The analytical modelling approach is used to describe the system mathematically and apply numerical methods to provide a general view of the system. As analytical results derive mainly from mathematical proofs, they are true as long as the conditions, parameters and assumptions are valid. Analytical modelling is less costly, more efficient and generally provides the best insight into the effects of various parameters and their interactions [63]. So, analytical modelling techniques, such as stochastic Petri nets and process algebra, have been used for the performance analysis of communication systems. However, the number of analytical studies of MANET is small [64]. This limitation of analytical modelling in MANET is achieved as a result of difficulty of incorporating the random mobility of nodes to these analyses which is the reason that most of these analytical studies suppose that the nodes are stationary. As it has already discussed in Chapter 2, nodes mobility is one of the major characteristic of the network that cannot be ignored which introduces the difficulty of using analytical modelling [65] to study the performance of this network.

When the system is rather large and complex, a mathematical formulation may not be feasible and easy to implement. Simulation introduces the solution for these environments. Simulation usually requires fewer simplification assumptions as almost all the details of the system specifications can be incorporated in a simulation model. On the other hand, simulation results are usually considered not as strong as the analytical results. Simulators are widely accepted as an efficient tool for studying the complex environment of networks in general. Also, it has proven to be a valuable tool to analyse system performance and examine the proposed model under different scenarios and conditions prior to the actual physical design of the system.

There is a wide variety of network simulation tools either commercial network simulators such as QualNet [66], OPNET [67] and MATLAB [68] or open-source network simulators such as Network Simulator (Version 2) NS-2 [69], OMNeT++

[70] and GloMoSim [71]. Each network simulator has its own strengths and weaknesses. The choice of the most appropriate one depends upon the following factors; the simulation platform, type of the simulation tool, and the user interfaces of the simulation tool [72].

Network simulation tools are examples of time-dependent simulation. This simulation approach maintains a simulation clock to keep track of simulation time and proceeds chronologically within events in the simulation. Time-dependent simulation can be classified into time-driven and event-driven simulation [62]. Time-driven simulation executes events at every fixed time interval of time units. On the other hand, an event-driven simulation executes events at any arbitrary time and does not proceed according to fixed time interval. It retrieves and removes an event with the smallest timestamp from the event list, executes it, and advances the simulation clock to the timestamp associated with the next event.

3.4.1 Selection Criteria

Network Simulator (Version 2) NS-2 [69] is the most widely used network simulation tool for networks researches. NS-2 is dominating most of the research in MANET and approximately 45% of published simulation-based MANET papers use NS-2 as a simulation tool [62]. It is an example of an event-driven simulation tool. NS-2 supports simulation of wired and wireless network functions and protocols. Due to its flexibility and modular nature, NS-2 has gained great popularity in the networking research community [62]. So, we select NS-2 as a simulation tool to analyse the behaviour of various MANET routing protocols in the presence of different attacks.

3.4.2 NS-2 Simulator

Network Simulator (Version 2) [69], widely known as NS-2, is an object-orientated discrete event simulator that was introduced by the VINT project at University of California in Berkeley in 1995. Later, NS2 was upgraded by the Monarch project at Carnegie Mellon University with support for node mobility. It is the most widely used open-source network simulator in the research field. It supports simulation

of wired and wireless network services and protocols and has the ability to study behaviours of both existing and new protocols.

One of the most important features of NS-2 is its ability to easily produce high number of randomly created scenarios for both traffic and movement patterns. NS2 has all the essential features like abstraction, visualisation, emulation, and traffic and scenario generation. NS-2 adopts a layered approach and supports a rich set of MANET routing protocols such as AODV, AOMDV, DSR, DSDV, as well as many others.

NS-2 consists of two programming languages, the first is C++ which defines the internal mechanism of the simulation and the second is Object-oriented Tool Command Language (OTcl) which is used to create and configure a network and acts as a user interface[62]. OTcl is an interpreted programming language which means changes in an OTcl file can be executed without compilation. NS2 is constructed to combine the advantages of these two languages as C++ is fast to run which makes it suitable for running the simulation. On the other hand, OTcl is fast to consider changes which makes it suitable for configuring the network. So, simulation components, their behaviour and the topologies are described by C++ code while OTcl scripts model the overall simulation behaviour and are used for binding.

3.4.3 Supporting Mobility in NS-2

The mobility model describes the mobile nodes movement patterns, looking at changes in their location, velocity and acceleration over time. Mobility pattern is an important factor that determines the protocol performance in different parameters. Mobility models [73] can be classified into three types; random walk, random waypoint and random direction mobility models. The random waypoint mobility model is the most commonly used mobility model for MANET research simulations and it is the one we have selected in our simulations [74].

3.4.4 Randomness in Scenarios Generation

NS2 supports deterministic and random mobility. Deterministic mobility is used in small networks and introduces an opportunity to control the movement of mobile nodes during simulation. Controlling the movement of nodes in a large network is hard to be achieved as long as the number of nodes increases in the network. On the other hand, although random mobility provides a good judgement on the behaviour of the network as it has wide range of tests, it does not provide the ability of reviewing or controlling the movement of mobile nodes in the network. NS2 provides an independent utility called *setdest* generated by CMU which creates movement-related OTcl statements using the random waypoint algorithm. These statements represent a fully random controllable movement of nodes and can be considered as deterministic mobility as it is available before simulation and can be included later in the simulation as a Tcl script. This utility is located in the directory *ns/indep-utils/cmu-scen-gen/setdest.*

NS2 also provides another independent utility called *cbrgen.tcl* that is written in Tcl to create traffic-related OTcl statements. This utility is a NS script and has to be included as well to the simulation script before starting the simulation. Despite the name, this utility can create both TCP and CBR traffic. This utility is located in the directory *ns/indep-utils/cmu-scen-gen/cbrgen.tcl.* The disadvantage of this available utility is that the connections between nodes are not fully randomised. The destination for a connection to node **n** is either node **n+1** for nodes that have ID smaller than 50 and **n+2** for nodes ID smaller than 75. So, we introduce a new modification to the existing utility that chooses a random destination for a connection to a source. For simulation purposes, the starting time of data sending for a connection is set randomly between 0 and 30 seconds. Figure 3.1 shows a flowchart of our modified utility while the detailed Tcl code is available in Appendix A.

We introduce a new independent utility called *nodes.tcl* that is written in Tcl as well to create node-related OTcl statements. This utility also is a NS script and has to be included to the simulation script before starting the simulation. This

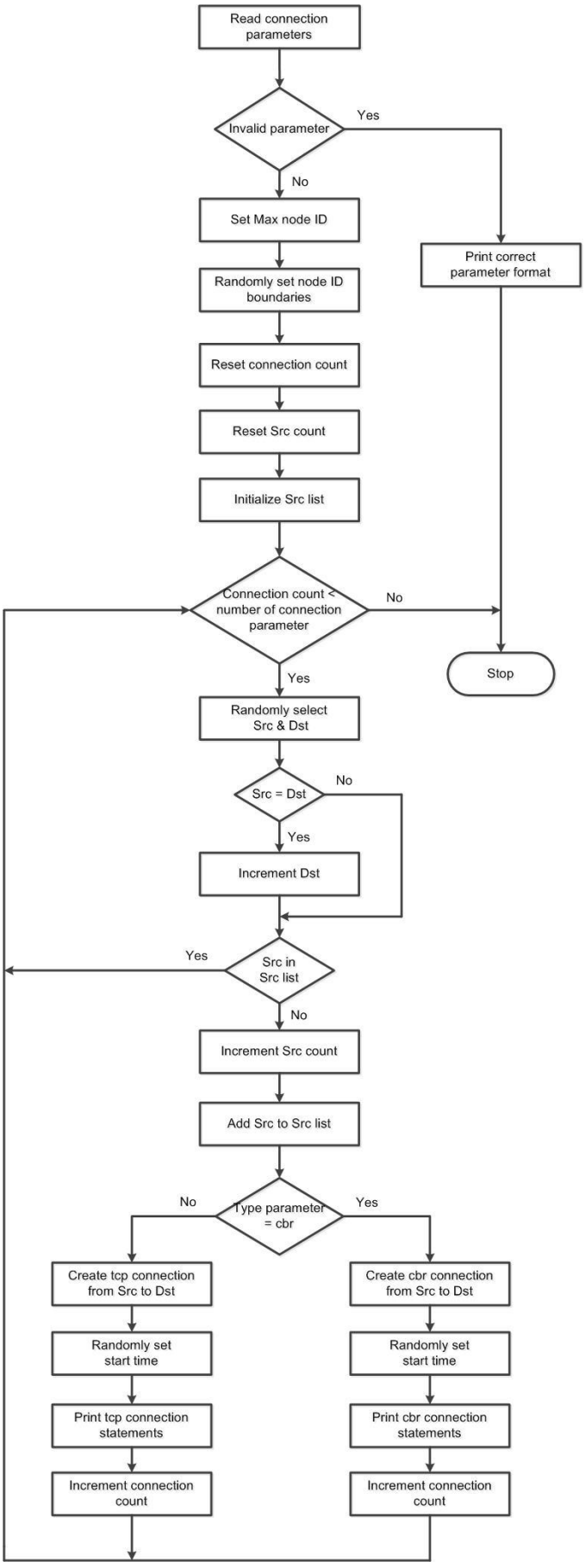

Figure 3.1: Modified Connection Generator

utility accepts from the user the number of nodes, the number of malicious nodes, the attack type, if the attack starting time has to be set randomly chosen or set to the starting time of simulation and optionally a seed value. The utility produces a Tcl script that randomly selects malicious nodes and the time they start their misbehaviour depends on the values predetermined by the user. Figure 3.2 shows a flowchart of our new *nodes.tcl* utility while the detailed Tcl code is available in Appendix B.

3.4.5 Adding Security to NS-2

NS-2 does not support security and cryptographic algorithms. So, we include one of the most popular libraries that implements security algorithms called OpenSSL. OpenSSL is a popular and effective open source toolkit implementing the Secure Sockets Layer (SSL) and Transport Layer Security (TLS), the most widely used protocols for secure network communications [75]. It is consists of two tools, SSL toolkit and a cryptography library. The SSL library provides an implementation of both the SSL and TLS protocols which can be used to secure applications that need to communicate over a network. SSL is currently the most widely deployed security protocol that is able to secure any protocol that works over TCP. The cryptography library provides the most popular algorithms for symmetric key and public key cryptography such as 3DES [38], AES [39] and RSA [42], as well as, hash algorithms, and message digests. It provides as well manipulating common certificate formats and managing and distributing key protocols. It includes also general-purpose helper libraries for pseudorandom number generator, manipulation of arbitrary precision numbers and buffers manipulations. OpenSSL is a cross-platform toolkit that works on both Unix and Windows and although its core library is written in C programming language, it can be used from C, C++ and other languages such as Python, Perl, and PHP.

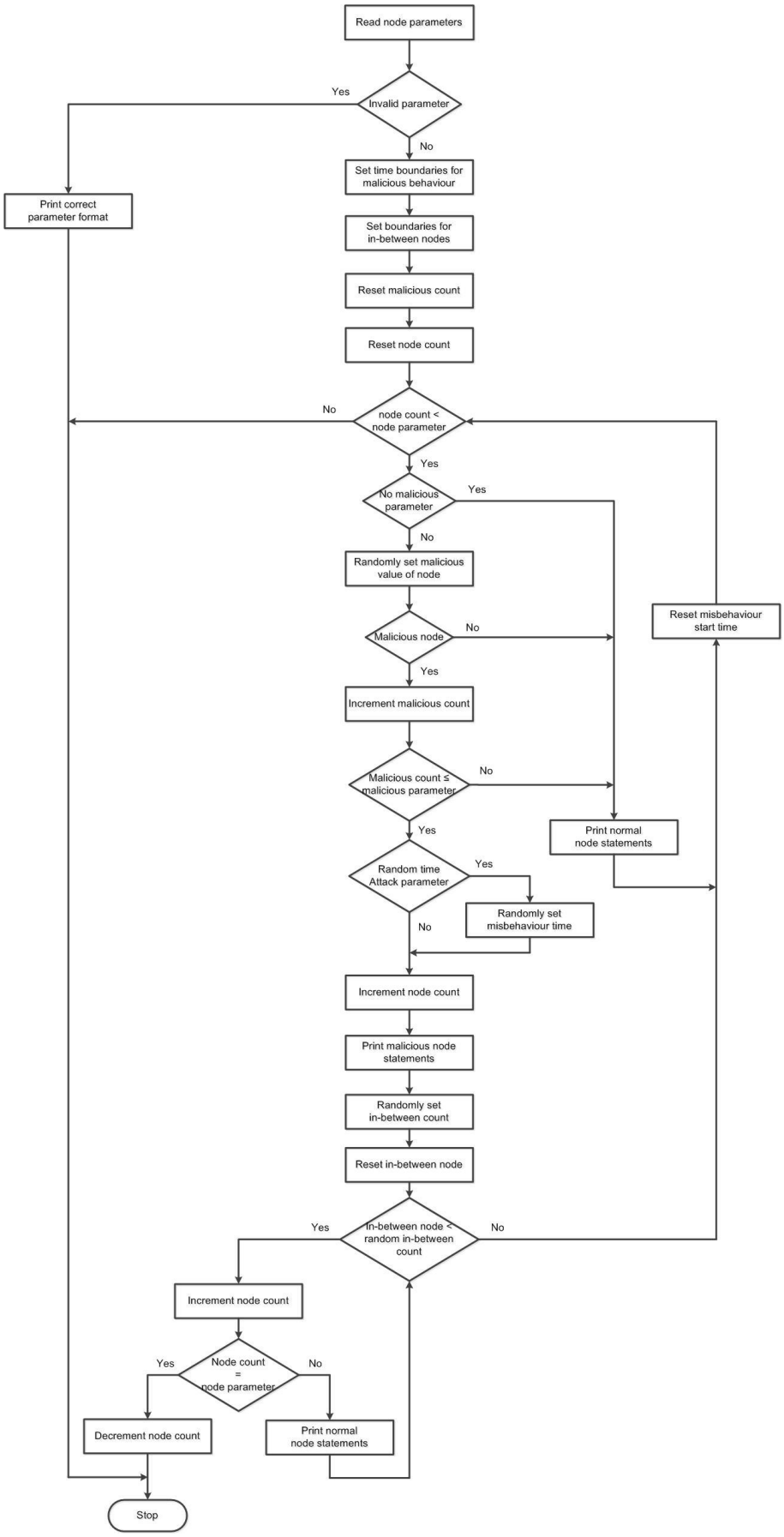

Figure 3.2: Malicious Node Scenario Generator

3.4.6 SAODV Implementation

Although NS-2 has many protocol implementations, it does not have an implementation for SAODV protocol. So, we introduce our own implementation that gives very similar results to those reported by the protocol author [76]. SAODV module is implemented by modifying the original AODV source code to include the security features such as hashing the hop count value of both RREQ and RREP packets and provide digital signature of these packets and RERR as well. These additional fields are appended to the RREQ, RREP and RERR packets of AODV protocol as a message extension. The size of the additional fields for RREQ, RREP, and RERR packets are 448 bytes, 448 bytes, and 404 bytes, respectively. RREQ and RREP include signature (64 bytes), top hash (16 bytes), hash (16 bytes) certificate (339 bytes) and other header information (13 bytes). RERR includes the signature (64 bytes) certificate (339 bytes) and other header information (1 byte). We include OpenSSL encryption library that includes a large number of different digital signatures and hashing algorithms to NS-2. We use Secure Hash Algorithm 1 (SHA-1) for generating and verifying the hash values of the hop count while we use Rivest-Shamir-Adleman (RSA) for signing and verifying the digital signatures of the routing packets.

Secure Hash Algorithm 1 (SHA-1) [47] is the most widely used SHA hash functions which is implemented in many widely used applications and protocols such as SSL, TLS, Secure Shell (SSH), and Internet Protocol Security (IPsec). SHA-1 produces a 160-bit long hash value as a message digest. In 2005, cryptanalysts found that attacks on SHA-1 achieve a result that the algorithm might not be secure enough which inspires most of the organisations to announce that they stop accepting SHA-1 certificates in SSL by 2017. SHA-2 family consists of six hash functions that have 224, 256, 384 or 512 bits long hash values which are used in SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256 functions respectively. Although SHA-2 has some similarity with SHA-1 algorithm, these attacks have not been successfully extended to SHA-2. In 2015, National Institute of Standards and Technology (NIST) has announced that SHA-3 has become a hashing standard. For

simplicity and achieve a fast simulation response, we use SHA-1 hash function to authenticate the hop count value for both the RREQ and RREP packets to ensure the integrity of the packet.

Rivest-Shamir-Adleman (RSA) [42] is public-key cryptography algorithm that is widely used for securing data over an insecure network such as the Internet. The security of RSA relies on the computational difficulty of factoring large integers. Although RSA keys are typically 1024-bits or 2048-bits long, it is expected that 1024-bits keys could be broken in the near future which directs governments and industries to approve 2048-bits long as the minimum key length. Again for simplicity and achieve a fast simulation response, we use 512-bits key for RSA digital signature in our simulation to authenticate the identity of the sender of the routing packet.

As it will be clarified in Chapter 4, SAODV has a high resistance to blackhole attack while its performance suffers a lot under flooding attack. So, we incorporate the flooding resistance solution to SAODV such as other MANET protocols to justify the effect of the suggested algorithms on its performance as it will be shown in Chapter 5. On the other hand, blackhole resisting mechanisms are not incorporated to SAODV as the original protocol is fully immune to this attack but it is used as a different solution to compare with as it will be shown in Chapter 6.

3.4.7 Simulation Approaches

The NS-2 simulator [69] is used to simulate the behaviour of various protocols in the presence of different attacks. Experiments have already been done on ***bwlf*** cluster of servers that consists of 32 machines. We use this large number of machines as the experiments have a very high number of scenarios. So, we distribute these scenarios between different machines to save the time of simulation and to overcome the storage size limitation that is caused by huge file size of each scenario as we will see in the section of simulation limitations. A machine in the ***bwlf*** cluster has 8 Intel processors 2.0 GHZ, 4 MB cache size, 12 GB RAM, and 1 TB hard disk size. NS-2 simulator is running under Linux CentOS 6.7 operating system. NS-2 is

working independent of the machine specifications as we are not focus on the time it takes to execute a scenario.

All experiments in the subsequent chapters are tested using simulation and the behaviour of these protocols with and without these new mechanisms are compared. Simulation results are obtained from 3 different movement scenarios, 3 different traffic scenarios and 3 different node-type (malicious or non-malicious) scenarios which mean that each metric value is the mean of the 27 runs. The node-type scenario is created randomly which implies that the addresses of malicious nodes and the starting time of the malicious behaviour are completely random. For simulation purposes, the starting time of the malicious node behaviour sets randomly between 0 and 50 seconds while the starting time of data sending for a connection regardless of the node-type sets randomly between 0 and 30 seconds. In all cases, the 90% confidence interval was small compared with the values being reported.

While our experiments are examined for both UDP and TCP traffic, the thesis is focused on the results of the TCP traffic only. While we examined our experiments for different numbers of nodes (25, 50, 75 and 100), only the case of 100 node networks is reported in the thesis which ensures a high density of nodes and gives malicious nodes a high number of neighbours. Node mobility was modelled with the random waypoint method. We examined our experiments on different node speeds (0, 5, 10, 15, 20, 25 and 30 m/s) and different simulation areas (500 m^2 and 1000 m^2) to ensure the effect of nodes' density on the results. For the consistency of the thesis, we reported only the larger area. As we will see in Chapter 4, the highest negative impact of malicious nodes usually appears on static networks and this effect decreases as node mobility increases [77], so most of the reported results in the thesis focus on the case of static networks.

All experiments in the subsequent chapters use the general simulation parameters that are shown in Table 3.1. Parameters used in Chapter 4 are slightly different from these are used in both Chapter 5 and Chapter 6 in simulation time and the number of connections in a scenario. In Chapter 4, as we will study the performance of MANET protocols under the different attacks, we choose a relatively small simulation time

Table 3.1: General Simulation Parameters

Simulation Area	1000 m^2
Number of Nodes	100
Number of Malicious Nodes	0 - 10
Node Speed	0 - 30 m/s
Pause Time	10 s
Traffic Type	TCP

of 180 seconds. On the other hand, we choose a large simulation time, 600 seconds, in both Chapter 5 and Chapter 6 to examine the ability of the suggested algorithms to discover the vast majority of malicious nodes especially for scenarios with a large number of malicious nodes. In addition, the number of connections per scenario is 70 in Chapter 4 while they are 150 in both Chapter 5 and Chapter 6. This large number of connections ensures that every node is involved in communication with at least one other partner during the simulation time to examine the ability of the suggested algorithms to work under the high density of connections.

3.4.8 Attacker Model

Attacker model for different types of attacks are different based on the misbehaviour action. We will examine the four attacks in Chapter 4 by simple attacker models. In Chapter 4, it will be clear that both flooding and blackhole attacks have dramatic negative effects on the performance of different MANET routing protocols. So, we will introduce intelligent attacker models in both Chapter 5 and Chapter 6 to examine the strength of the suggested algorithms to resist flooding and blackhole attacks respectively.

The selfish attack model assumes that a selfish node follows the normal protocol behaviour only if it requires to send data. Otherwise, it drops the received packet whether it is data or routing packet and whether it is for itself or it should be forwarded to another node.

The grayhole attack model assumes that a grayhole node follows the normal protocol behaviour in reacting to a RREQ. If the node is used later to forward data, it drops data although it has a fresh route to the destination and it has already agreed to cooperate by previously sending a true RREP.

While the flooding attack model is simple in Chapter 4, a more intelligent attacker is assumed in Chapter 5 to examine the strength of the suggested algorithms. A malicious node periodically broadcasts a fake RREQ every 0.5 second in Chapter 4 while it randomly sets the interval between each two successive RREQ in Chapter 5. The simple attacker model sets the hop count of the RREQ to 1 and the TTL value to NETWORK_DIAMETER which is the highest value to ensure that widespread of the RREQ to the maximum possible number of nodes in the network. On the other hand, the more intelligent attacker randomly sets the generated hop count between 2 and 4. The value 2 ensures that a malicious node spoofs its neighbours that it forwards a RREQ received from another node, while the value 4 ensures that this RREQ travels at least (NETWORK_DIAMETER - 4) hops to flood the network.

The blackhole attack model also is simple in Chapter 4 while it assumes an intelligent attacker in Chapter 6 to examine the strength of the suggested algorithms. In Chapter 4, it assumes that once a blackhole node receives a RREQ, it unicasts a fake RREP with a hop count to 2 to spoof other nodes about best route; i.e. 1 hop count only from the RREQ destination. In Chapter 6, the attacker constructs a fake RREP that includes a randomly generated hop count between 2 and 4 to spoof other nodes about best route; i.e. 1 to 3 hop counts only from the RREQ source. Both attacker models assign the destination sequence number value of this fake RREP as equal to the received one in the RREQ plus a randomly generated number between 10 and 30 to spoof other nodes about the freshness of this RREP. A malicious node initiating a blackhole attack generates a fake RREP for each RREQ it receives to incorporate itself in all routes, therefore all packets are sent to a point where they are not forwarded anywhere which is a form of a DoS attack. Later, when a source node uses this malicious node to forward data, it drops the received data.

3.4.9 Simulation Limitations

Our experiments have been conducted on a wide range of environments such as multiple simulation areas, simulation times, number of nodes in the networks, number of

malicious nodes in the networks, starting time of misbehaving for malicious nodes, traffic type, and number of connections in a scenario and starting time of transmission for a connection. Although this wide variety of parameters introduces a trust in the results, this work has a number of limitations that has to be considered in the future work. The main limitations are as follows:

1. Evaluating the existing protocols and our new mechanisms have been achieved for networks that have at most 100 nodes. So, we did not examine our work in larger networks. This is because the inflation of the capacity of trace file that is produced by a scenario as a result of malicious nodes behaviour. This inflation is proportional to the number of malicious nodes in the network. As an example, a network that contains 100 nodes produces an approximately 1.20 GBytes trace file if the network has no malicious nodes while this trace file is inflated to an approximately 4.25 GBytes if the number of malicious nodes increases to 10 especially for high mobility networks. As we mentioned earlier, these simulation results are obtained from 3 different movement scenarios, 3 different traffic scenarios and 3 different node-type (malicious or non-malicious) scenarios which mean that each metric value is the mean of the 27 runs to achieve a confidence interval. This means that analysing behaviour of network for each malicious count requires 27 times the size of a scenario. Testing networks that contain 200 nodes generates a file size of an approximately 7.30 GBytes which introduces a difficulty to test larger networks. In addition, the time to execute a scenario is proportional to the number of malicious nodes in the network. Sometimes, it takes more than 30 minutes to execute a scenario which adds another difficulty to evaluate the larger networks.

2. Although our mechanism to resist the blackhole attacks succeeded in discovering and excluding only genuine malicious nodes, we did not achieve the same success in resisting flooding attacks. Flooding Attack Resisting Mechanism (FARM) succeeded in detecting an excluding more than 80% of malicious neighbours in the simulation time with a highly trusted ratio ex-

ceeds than 90% if the algorithm is incorporated to AODV, DSR or SAODV. On the other hand, FARM achieves a smaller success in AOMDV with a trusted ratio approximately 40%.

3.4.10 Evaluation Metrics

There are several metrics used to evaluate a network performance. However for the purposes of this research, we focus on the following metrics to examine the performance of various protocols in the presence of different attacks.

Packet Delivery Ratio (PDR): The ratio of data packets that are successfully delivered to destinations compared to the number of data packets that have been sent out by sources.

Throughput: The number of data bits delivered to the application layer of destination nodes in unit time measured in bps.

End-to-End Delay (EED): The average time taken for data packets to be transmitted across the network from sources to destinations.

Routing Overhead: The number of routing packets for route discovery and route maintenance needed to deliver the data packets from sources to destinations. When we compare two or more different protocols, we use the size of the routing packets measured in KBytes to consider the differences between the routing packet sizes of theses protocols

Normalized Routing Load (NRL): The ratio of the number of transmitted routing packets to the number of received data packets.

Route Discovery Latency (RDL): The average of the delays between sending RREQs from a source and receiving the first corresponding RREP.

True Exclusion Ratio: The ratio of successful exclusion of genuine malicious nodes to the total number of exclusions.

Total Exclusions: The total number of neighbour exclusions during the simulation time.

Malicious Discovery Ratio: The ratio of malicious nodes discovered as time progresses to the total number of malicious nodes that should be discovered.

3.5 Summary

MANET routing protocols are designed based on the assumption that all nodes cooperate without maliciously disrupting the operation of the routing protocol. Reactive MANET protocols are initiated by a source whenever it requires to send data to a destination node. AODV, DSR, AOMDV, and SAODV routing protocols are selected to study their behaviours under various attacks. Both AODV and DSR are well-known reactive routing protocols in MANET while AOMDV represents multipath routing protocols and SAODV represents secured routing protocols in MANET. Active attacks are classified based on modification, impersonation or fabrication to the routing packets. Flooding, blackhole, grayhole and selfish attacks are selected to represent the wide range of malicious behaviours and analyse the effect of them on a network performance. Flooding and blackhole attacks represent the fabrication attacks that can launch a denial of service. Selfish attack represents the loss of cooperation that MANET design is based. Grayhole attack represents partial misbehaviour of a malicious node in complying with the protocol operation. NS-2 is the most widely used network simulation tool for networks researches. NS-2 has been selected to study the performance of different protocols in the presence of various attacks.

Chapter 4

Routing Protocols under Attacks

4.1 Introduction

In this chapter, we use NS-2 to study the performance of AODV, DSR, AOMDV and SAODV routing protocols in the presence of flooding, blackhole, grayhole and selfish attacks. Results from these simulations have been published in [77], [78], [76] and [79].

The rest of the chapter is organized as follows. In Section 4.2, a simulation approach is presented. In Section 4.3, the impact of some attacks on AODV is discussed. In Section 4.4, DSR performance under these attacks is introduced. In Section 4.5, the impact of these attacks on AOMDV is discussed. In Section 4.6, SAODV performance in the presence of these attacks is introduced. In Section 4.7, the performance comparison of these protocols is discussed. In Section 4.8, summary is presented.

4.2 Simulation Approach

NS-2 simulator [69] is used to simulate flooding, blackhole, grayhole and selfish attacks. The simulation is used to analyse the performance of AODV, DSR, AOMDV and SAODV routing protocols under these attacks. The parameters used are shown in Table 4.1. While we examined these protocols on both UDP and TCP traffic, the chapter is focused on the results of the proposed mechanism on the TCP traffic

only. We examine these protocols for different number of nodes (25, 50, 75 and 100) and different node speeds (0, 5, 10, 15, 20, 25 and 30 m/s) although we present the results for (0, 10, 20 and 30 m/s) for clarity. Similarly, only the case of 100 node networks is reported, corresponding to a high density of nodes. This gives malicious nodes a high number of neighbours.

Our flooding attack model assumes that a malicious node periodically broadcasts a fake RREQ every 0.5 second. The malicious node randomly chooses a destination address between 200 and 300 with a randomly generated destination sequence number. It sets the hop count of the RREQ to 1 and the TTL value to NETWORK_DIAMETER which is the highest value to ensure that widespread of the RREQ to the maximum possible number of nodes in the network.

Our blackhole attack model assumes that once a blackhole node receives a RREQ, it unicasts a fake RREP without reference to its routing table. It sets the hop count to 2 to spoof other nodes about best route; i.e. 1 hop count only from the RREQ destination. The attacker assigns the destination sequence number value of this fake RREP as equal to the received one in the RREQ plus a randomly generated number between 10 and 30 to spoof other nodes about the freshness of this RREP. A malicious node initiating a blackhole attack generates a fake RREP for each RREQ it receives to incorporate itself in all routes, therefore all packets are sent to a point where they are not forwarded anywhere which is a form of a DoS attack. Later, when a source node uses this malicious node to forward data, it drops the received data.

Our grayhole attack model assumes that a grayhole node follows the normal protocol behaviour in reacting to a RREQ. If the node is used later to forward data, it drops data although it has a fresh route to the destination and it has already agreed to cooperate by previously sending a true RREP.

Our selfish attack model assumes that a selfish node follows the normal protocol behaviour only if it requires to send data. Otherwise, it drops the received packet whether it is data or routing packet and whether it is for itself or it should be forwarded to another node.

Table 4.1: Reactive Protocols under Attacks Simulation Parameters

Simulation Time	180 s
Simulation Area	1000 m^2
Number of Nodes	100
Number of Connections	70
Number of Malicious Nodes	0 - 10
Node Speed	0 - 30 m/s
Pause Time	10 s
Traffic Type	TCP

Generally, regardless of the number of malicious nodes that joins the network, static networks achieve better performance than mobile networks.

4.3 AODV under Attacks

AODV has a huge degradation of its performance under both flooding and blackhole attacks. On the other hand, selfish and grayhole attacks have a small negative impact on their performance. Details of the simulations are presented in the following sections.

4.3.1 AODV under Flooding Attack

The results show that the flooding attack has a severe impact on AODV performance. The effect of flooding attack on the packet delivery ratio is shown in Figure 4.1. Results show that the flooding attack has a negative impact on the PDR of AODV. This negative impact is higher in high node mobility networks than in static networks.

Figure 4.2 shows the effect of flooding attack on the network throughput. Throughput of AODV decreases dramatically as the number of malicious nodes increases. The network throughput is more highly affected in static networks than in high node mobility networks.

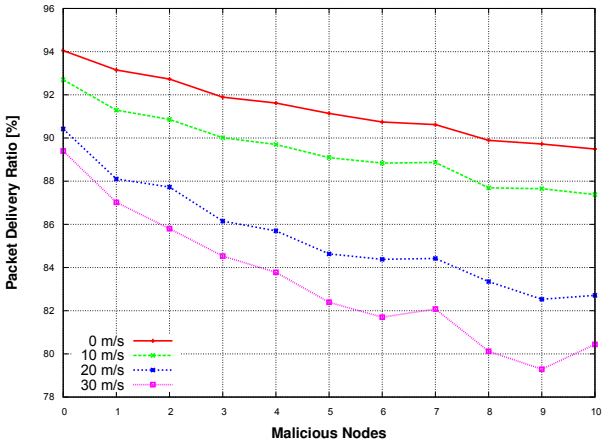

Figure 4.1: AODV Packet Delivery Ratio under Flooding

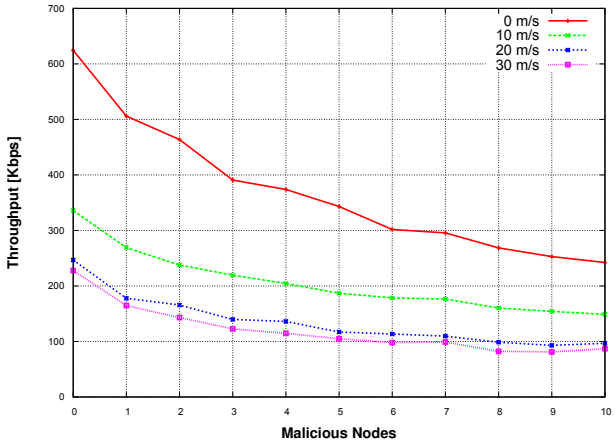

Figure 4.2: AODV Network Throughput under Flooding

The effect of flooding attack on the end-end-delay is shown in Figure 4.3. The result shows that the delay increases as the number of malicious nodes increases regardless of node mobility.

Figure 4.4 shows the effect of flooding attack on the routing overhead. Routing overhead increases as the number of malicious nodes increases regardless of the speed of the nodes.

4.3.2 AODV under Blackhole Attack

A blackhole attack has a large impact on the AODV performance. The effect of the blackhole attack on the packet delivery ratio is shown in Figure 4.5. The PDR

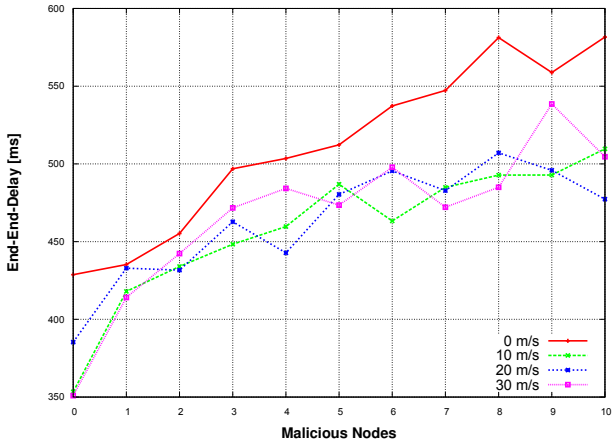

Figure 4.3: AODV End-End-Delay under Flooding

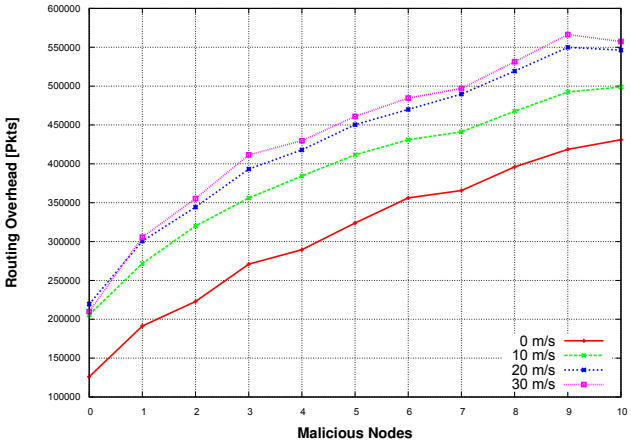

Figure 4.4: AODV Routing Overhead under Flooding

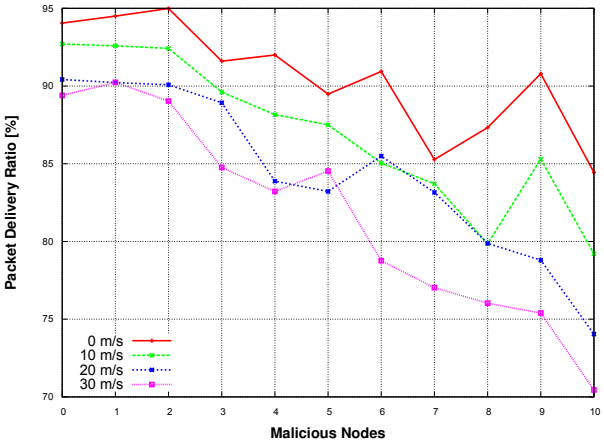

Figure 4.5: AODV Packet Delivery Ratio under Blackhole

of AODV decreases dramatically as the number of malicious nodes increases. This performance degradation is more pronounced in higher nodes speed.

Figure 4.6 shows the effect of blackhole attack on the network throughput. The throughput of AODV decreases dramatically as the number of malicious nodes increases. The network throughput is more highly affected in static networks than in networks with high node mobility.

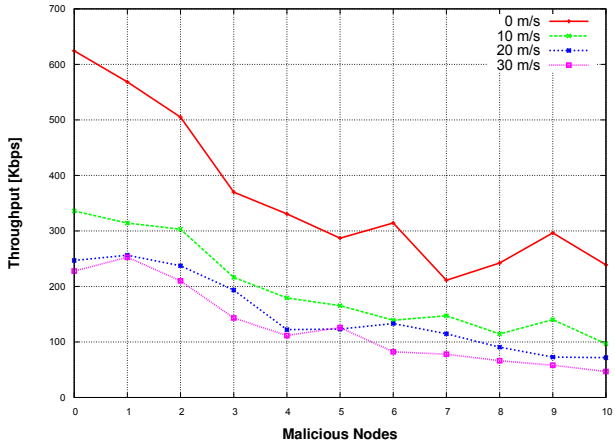

Figure 4.6: AODV Network Throughput under Blackhole

The effect of blackhole attack on the end-end-delay is shown in Figure 4.7. An expected logical result has to be increasing the EED as the number of malicious nodes increases and this delay should be at least equal to the delay of the network in the absence of malicious nodes. However, the results show that the delay of AODV is reduced as the number of malicious nodes increases regardless of the node mobility. This result is slightly paradoxical as the attack improves the delay. This is a misleading result because the delay is only measured on packets that reach their destinations and since the blackhole nodes drop all the received data, the number of packets that will be considered in calculating the delay decreases as the number of malicious nodes increases. The routes that avoid blackhole nodes suffer less competition, and hence reduced delay.

Figure 4.8 shows the effect of blackhole attack on the routing overhead. The routing overhead of AODV decreases dramatically as a result of malicious nodes especially for the first three malicious nodes regardless of the node mobility. These results also are slightly confusing as the blackhole attack improves the routing over-

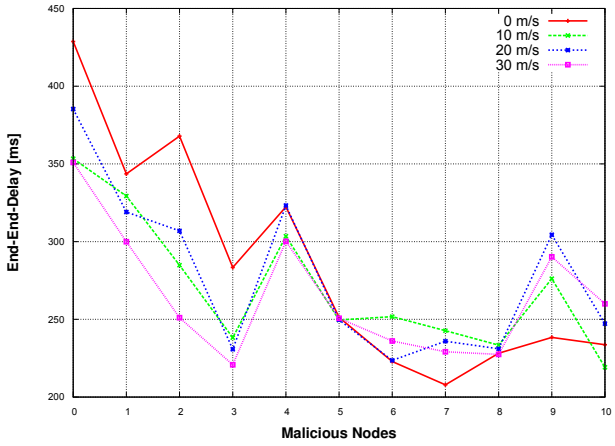

Figure 4.7: AODV End-End-Delay under Blackhole

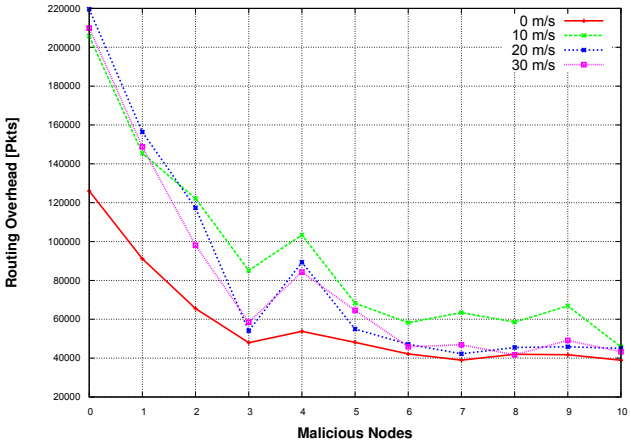

Figure 4.8: AODV Routing Overhead under Blackhole

head. This is because the blackhole nodes stop rebroadcasting the RREQ which decreases the total number of RREQ packets, one of factors used to measure the routing overhead.

4.3.3 AODV under Grayhole Attack

The results show that the grayhole attack has no significant effect on the AODV performance. The effect of grayhole attack on the packet delivery ratio is shown in Figure 4.9. While the PDR of AODV is constant regardless of the number of malicious nodes, the PDR is slightly better in static networks than in high node mobility networks.

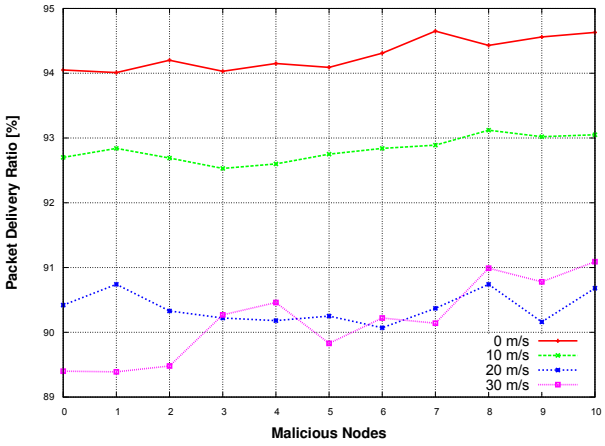

Figure 4.9: AODV Packet Delivery Ratio under Grayhole

Figure 4.10 shows the effect of grayhole attack on the routing overhead. The routing overhead of AODV decreases slightly as the number of malicious nodes increases regardless of the node mobility. This result is also paradoxical as the grayhole attack improves the routing overhead. The reason, as discussed before in the blackhole attack, is that the grayhole nodes stop rebroadcasting the RREQ which decreases the routing overhead.

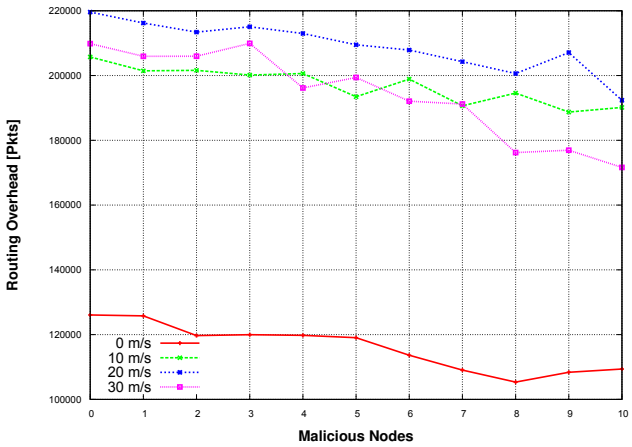

Figure 4.10: AODV Routing Overhead under Grayhole

4.3.4 AODV under Selfish Attack

The results show that the selfish attack has no significant effect on the AODV performance. As the grayhole node drops all data packets and the selfish node drops all data and routing packets, the grayhole attack simulation produces very

similar results to the selfish attack. This is because the metrics are calculated based on the received data packets which are very similar for both attacks. So, we do not include the results of the selfish attack on AODV.

4.4 DSR under Attacks

DSR has a dramatic collapse of its performance under flooding attack and a smaller negative impact under blackhole attack. On the other hand, selfish and grayhole attacks have no significant impact on its performance. Details of the simulations are presented in the following sections.

4.4.1 DSR under Flooding Attack

The results show that the flooding attack has a severe impact on the DSR performance. Simulation shows that the flooding attack has very similar effect on both AODV and DSR. Figure 4.11 and Figure 4.12 show the effect of flooding attack on the network throughput and overhead respectively which clarifies the performance similarities of both protocols.

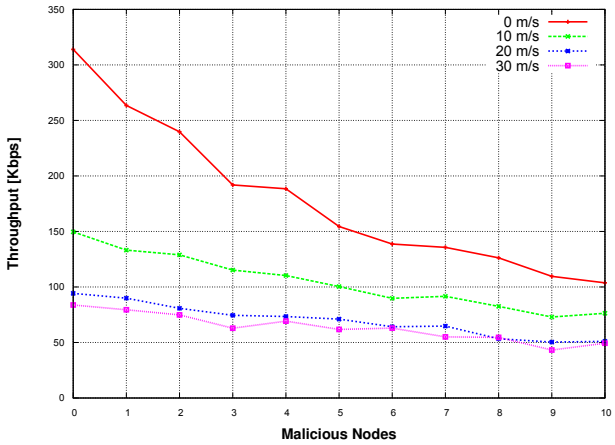

Figure 4.11: DSR Network Throughput under Flooding

4.4.2 DSR under Blackhole Attack

The results show that the blackhole attack has a small impact on the DSR performance. The effect of blackhole attack on the packet delivery ratio is shown

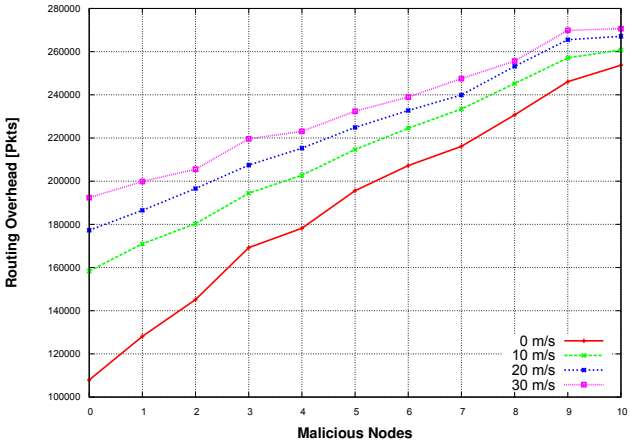

Figure 4.12: DSR Routing Overhead under Flooding

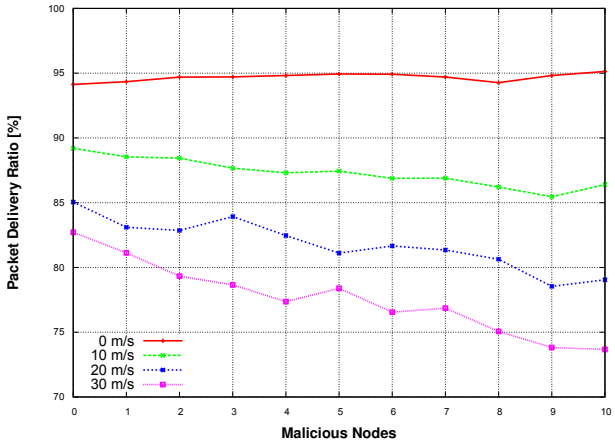

Figure 4.13: DSR Packet Delivery Ratio under Blackhole

in Figure 4.13. The PDR of DSR is nearly constant for static nodes and the performance degradation increases as the nodes' speed increases.

The effect of blackhole attack on the end-end-delay is shown in Figure 4.14. While the average delay is approximately constant for mobile nodes, when the network is static the delay for DSR is reduced as the number of malicious nodes increases. The reason of this paradoxical result of statics nodes has been discussed before in AODV as the number of data packets that is considered in calculating the delay decreases as the number of malicious nodes increases.

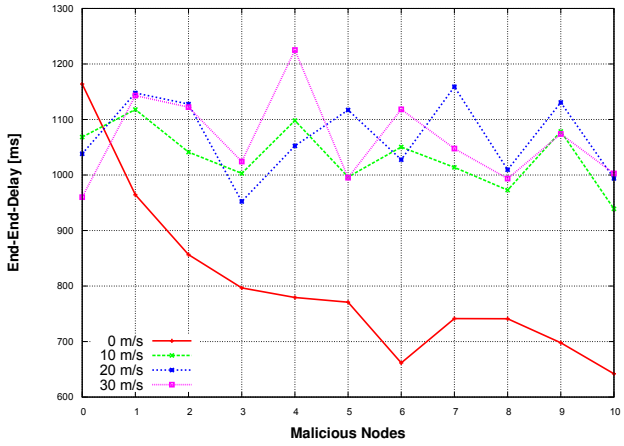

Figure 4.14: DSR End-End-Delay under Blackhole

Figure 4.15 shows the effect of blackhole attack on the routing overhead. While the average routing overhead of DSR is constant for static network, it increases slightly as a result of malicious nodes.

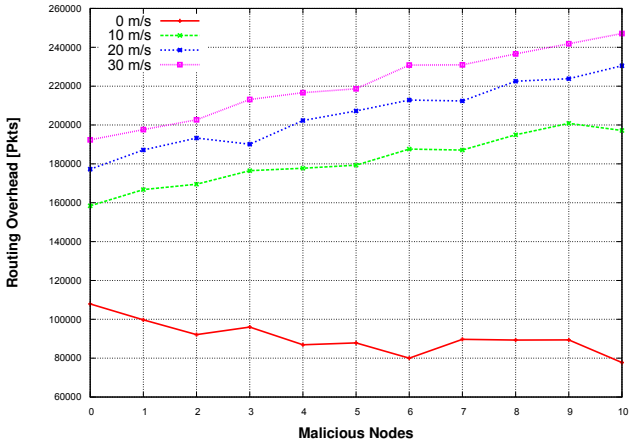

Figure 4.15: DSR Routing Overhead under Blackhole

4.4.3 DSR under Selfish Attack

The results show that the selfish attack has no significant effect on the DSR performance. Packet delivery ratio and network throughput are nearly constant regardless of node mobility and the number of malicious nodes. The effect of selfish attack on delay is shown in Figure 4.16. The average delay of DSR is constant for high speed nodes while it decreases as the number of malicious nodes increases for static and low speed nodes.

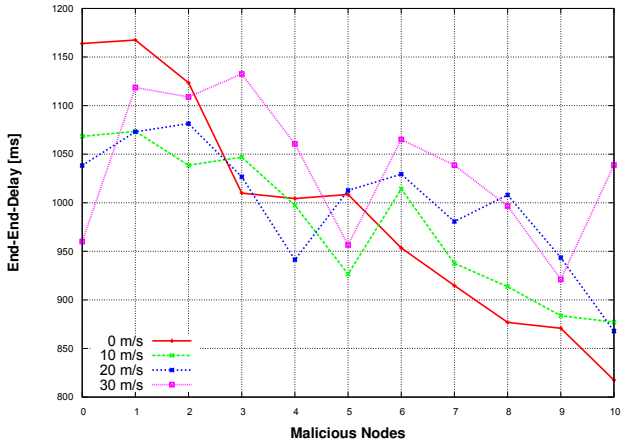

Figure 4.16: DSR End-End-Delay under Selfish

Figure 4.17 shows the effect of selfish attack on the routing overhead. The routing overhead decreases slightly as a result of malicious nodes regardless of node mobility.

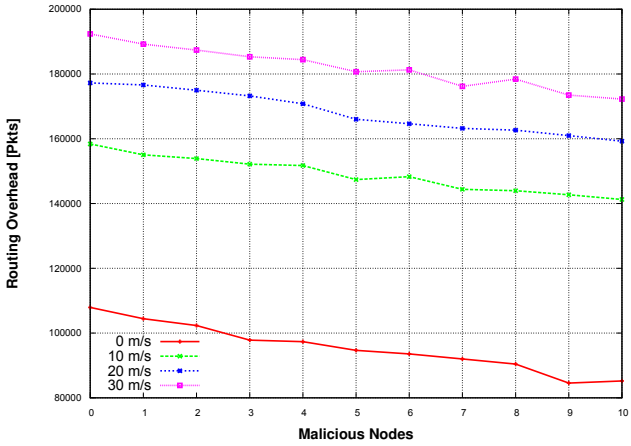

Figure 4.17: DSR Routing Overhead under Selfish

4.4.4 DSR under Grayhole Attack

As mentioned earlier, the grayhole and the selfish attacks achieve very similar results. This is because, as we discussed previously, the metrics are calculated based on the received data packets which are same for both attacks. The major difference between the performance of DSR under the two attacks is that while the routing overhead of selfish attack slightly decreases as the number of malicious nodes increases, its value is nearly constant for grayhole attack. This is because the selfish nodes drop the

routing packets in addition to the data packets that are dropped by the grayhole nodes. So, we do not include the results of the grayhole attack on DSR.

4.5 AOMDV under Attacks

AOMDV has a small degradation on its performance under various attacks. Details of the simulations are presented in the following sections.

4.5.1 AOMDV under Flooding Attack

The results show that the flooding attack has a severe impact on the AOMDV performance especially for static networks. Figure 4.18 shows the effect of flooding attack on the network throughput. Throughput of AOMDV decreases dramatically as the number of malicious nodes increases for static nodes. The higher the nodes speed the lower the effect of the flooding attack on the network throughput.

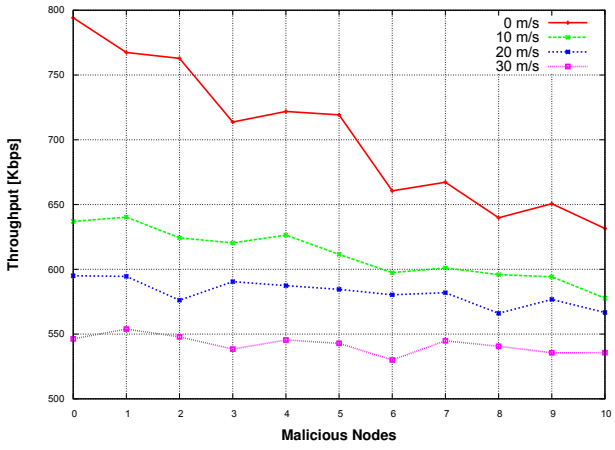

Figure 4.18: AOMDV Network Throughput under Flooding

Figure 4.19 shows the effect of flooding attack on the routing overhead. The result shows that this attack has a dramatic impact on the routing overhead especially for static nodes. This effect decreases as the node mobility increases.

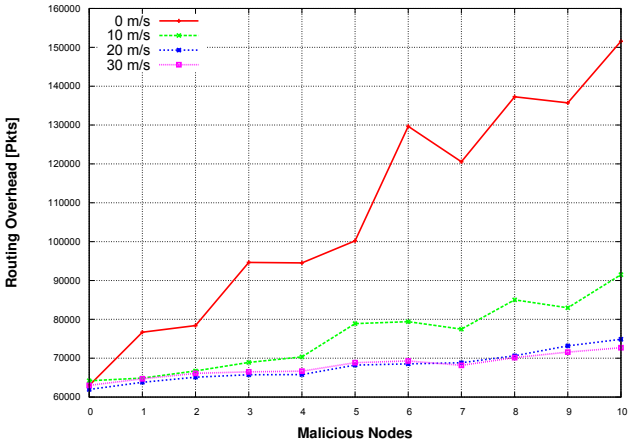

Figure 4.19: AOMDV Routing Overhead under Flooding

4.5.2 AOMDV under Blackhole Attack

The results show that the blackhole attack has a small impact on the AOMDV performance. Figure 4.20 shows that the blackhole attack has a little degradation on the network throughput regardless of the nodes speed.

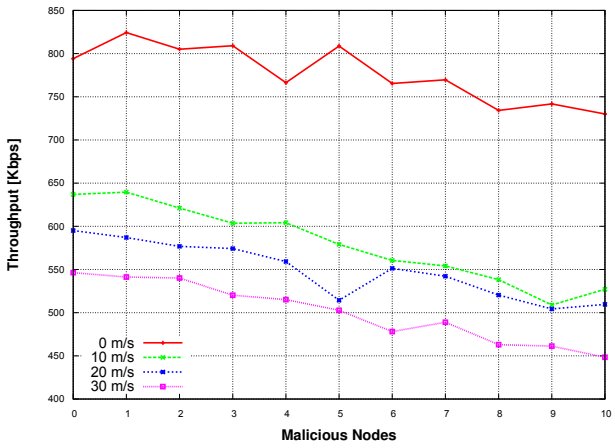

Figure 4.20: AOMDV Network Throughput under Blackhole

The effect of blackhole attack on the end-end-delay is shown in Figure 4.21. While the average delay of AOMDV is approximately constant for mobile networks, its value decreases as the number of malicious nodes increases for the reason that has been mentioned earlier in AODV.

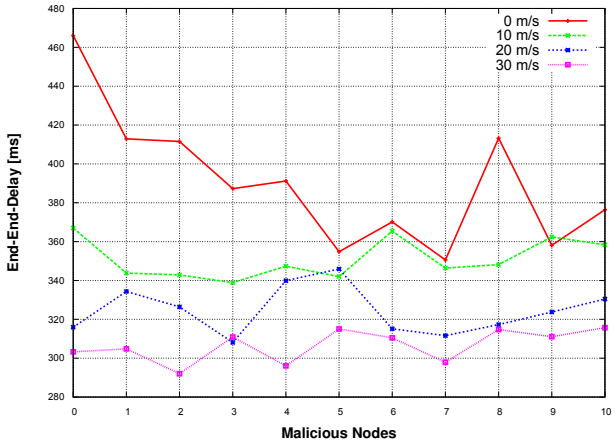

Figure 4.21: AOMDV End-End-Delay under Blackhole

Figure 4.22 shows the effect of blackhole attack on the routing overhead. The routing overhead of AOMDV increases dramatically as a result of malicious nodes especially for mobile networks. The results show that this increase is small for static networks.

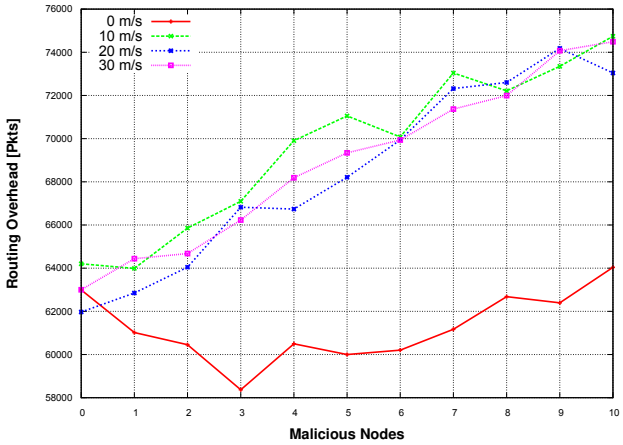

Figure 4.22: AOMDV Routing Overhead under Blackhole

4.5.3 AOMDV under Grayhole Attack

The results show that the grayhole attack has no significant effect on the AOMDV performance. Figure 4.23 shows the effect of grayhole attack on the routing overhead. The routing overhead of AOMDV decreases slightly as the number of malicious nodes increases. The reason has been discussed earlier in AODV.

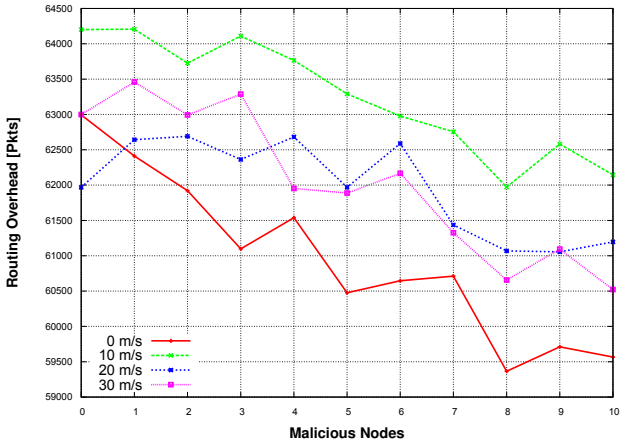

Figure 4.23: AOMDV Routing Overhead under Grayhole

4.5.4 AOMDV under Selfish Attack

As mentioned in both AODV and DSR, the grayhole and the selfish attacks achieve very similar results. The results show that the selfish attack has no significant effect on the AOMDV performance. So, we do not include the results of the selfish attack on AOMDV.

4.6 SAODV under Attacks

While SAODV has a dramatic collapse on its performance under flooding attack, it has a high resistance to blackhole, selfish and grayhole attacks. SAODV does not forward the routing packets without ensuring authenticity and integrity which explains its success in resisting blackhole, grayhole and selfish attacks. SAODV cannot resist the flooding attack because a malicious node impersonates a non-existent node which could not be discovered by other non-malicious nodes. Details of the simulations are presented in the following sections.

4.6.1 SAODV under Flooding Attack

The results show that the flooding attack has a severe impact on the SAODV performance. The effect of flooding attack on the packet delivery ratio is shown in Figure

4.24. The flooding attack has a highly negative impact on the PDR of SAODV. This negative impact is independent of the node mobility.

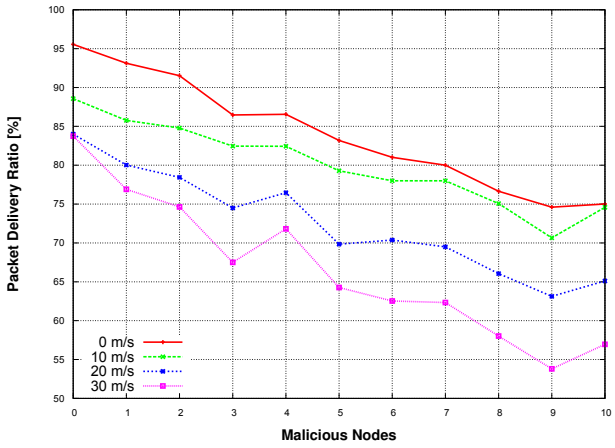

Figure 4.24: SAODV Packet Delivery Ratio under Flooding

Figure 4.25 shows the effect of flooding attack on the network throughput. Throughput of SAODV decreases dramatically as the number of malicious nodes increases. The reduction in the throughput is most dramatic in static networks.

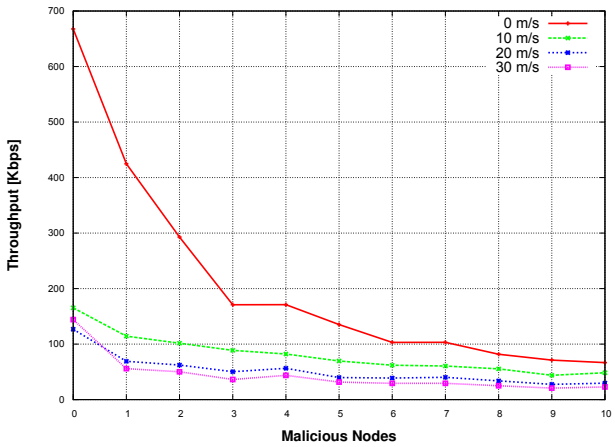

Figure 4.25: SAODV Network Throughput under Flooding

The effect of flooding attack on the EED is shown in Figure 4.26. The result shows that the delay increases as the number of malicious nodes increases regardless of node mobility. Static networks have a higher impact on the delay than mobile networks.

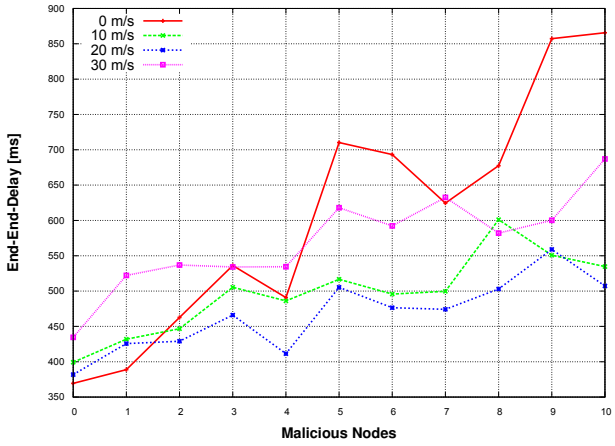

Figure 4.26: SAODV End-End-Delay under Flooding

Figure 4.27 shows the effect of flooding attack on the routing overhead. The routing overhead increases as the number of malicious nodes increases regardless of the speed of the nodes. The effect is smaller in mobile networks than in static networks.

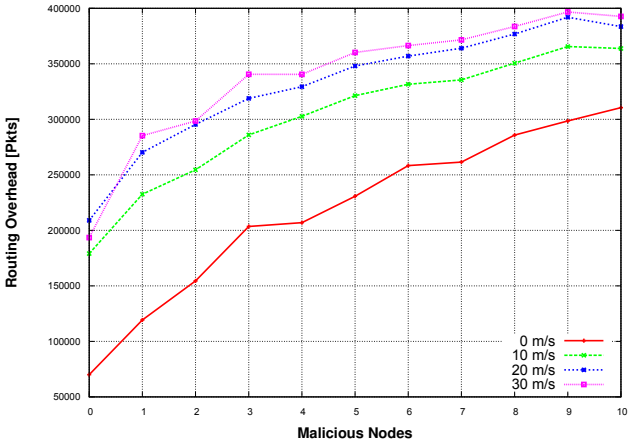

Figure 4.27: SAODV Routing Overhead under Flooding

SAODV is highly resistant to all other attacks. Verifying the signature of a routing packet using the sender's public key before rebroadcasting is a highly effective way to discard malicious nodes packets. Results of various attacks show that the network performance is independent of the number of malicious nodes joining the network. Figure 4.28 and Figure 4.29 show the effect of blackhole attack on the network throughput and the routing overhead respectively as an example to its high

resistance to different attacks. Figure 4.28 shows that the throughput of SAODV does not change significantly in the presence of malicious nodes.

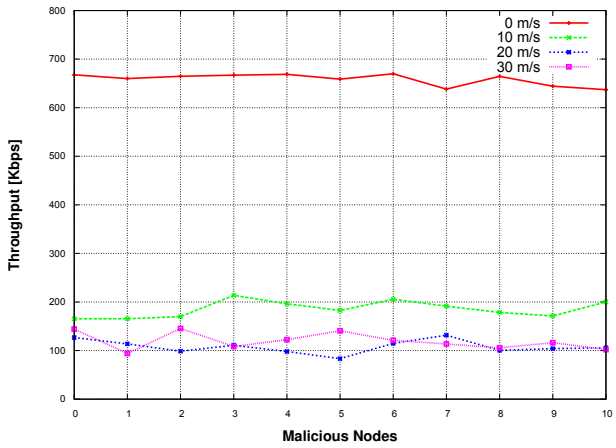

Figure 4.28: SAODV Network Throughput under Blackhole

Similarly, Figure 4.29 shows that the average routing overhead is not affected by the number of malicious nodes joining the network.

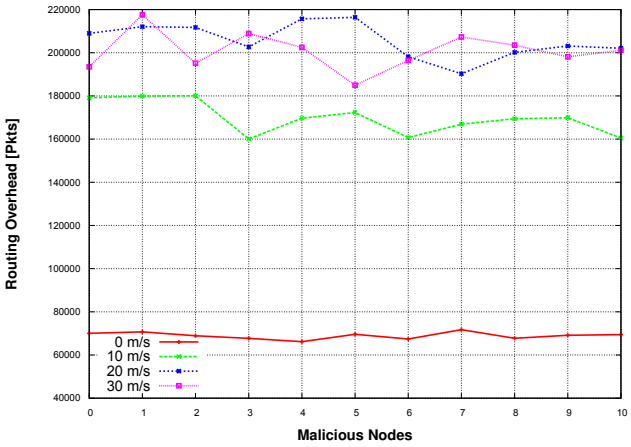

Figure 4.29: SAODV Routing Overhead under Blackhole

4.7 Performance Comparison

In the previous sections, we have studied various protocols individually under different attacks. From the above discussions, we conclude that the highest negative impact of malicious nodes usually appears on static networks and this effect decreases as node mobility increases [77]. In this section, we compare the performance

of various protocols in static networks in the presence of different attacks to determine the discrepancies between them. As the routing packet sizes are different from a protocol to another, we consider the routing overhead as the size of the routing packets measured in KBytes.

4.7.1 Flooding Attack

Figure 4.30 shows the network throughput of the examined protocols under the flooding attack. The flooding attack has a high negative impact on both AODV and SAODV while DSR and AOMDV suffer less from the attack. SAODV has the most dramatic degradation of its performance under the flooding attack. While AOMDV achieves the best performance, the DSR achieves the worst network throughput in the presence of the flooding attack.

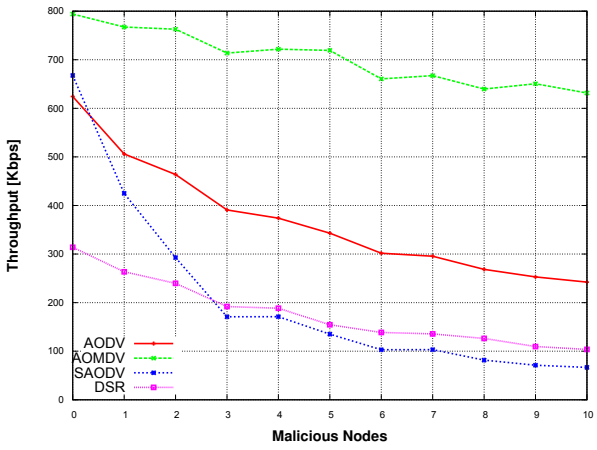

Figure 4.30: Network Throughput under Flooding

Routing overhead of the examined protocols under the flooding attack is shown in Figure 4.31. The routing overhead of both SAODV and DSR is inflated as the number of malicious nodes increases, the routing overhead of AODV inflates with a smaller ratio and the effect on AOMDV is ever smaller. While AOMDV achieves the best overhead, the SAODV achieves the worst value under the flooding attack.

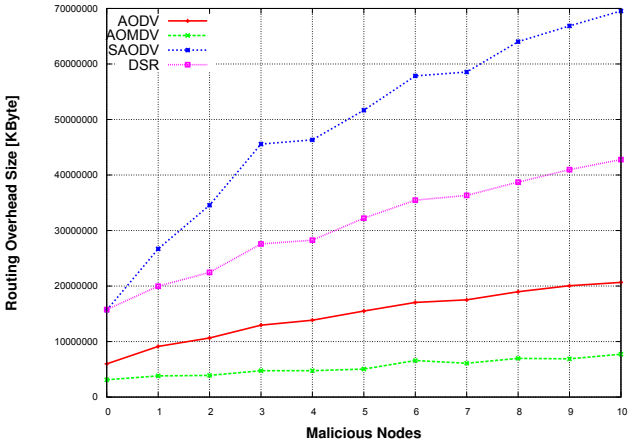

Figure 4.31: Routing Overhead under Flooding

4.7.2 Blackhole Attack

Figure 4.32 shows the network throughput of the examined protocols under the blackhole attack. The figure shows that AODV is the only protocol that has huge degradation on its throughput as a result of the blackhole attack. While AOMDV achieves the best throughput, DSR is the worst performing protocol especially when the number of malicious nodes is small.

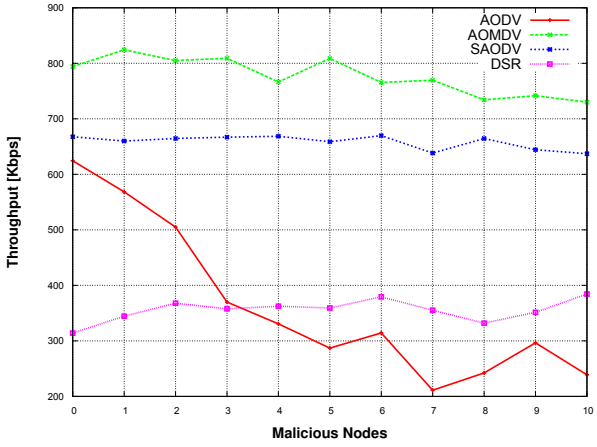

Figure 4.32: Network Throughput under Blackhole

Routing overhead of the examined protocols under the blackhole attack is shown in Figure 4.33. While both AOMDV and SAODV achieve an approximately constant overhead, the overhead of the AODV and DSR decreases as a result of this attack. This confusing result has been mentioned earlier in AODV analysis. While

AOMDV achieves the lowest overhead, SAODV's overhead is the largest, almost independently of the number of malicious nodes.

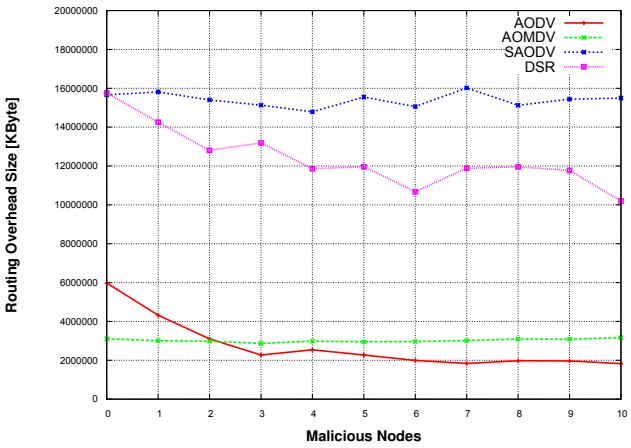

Figure 4.33: Routing Overhead under Blackhole

4.7.3 Selfish Attack

Figure 4.34 shows the network throughput of the examined protocols under the selfish attack. The selfish attack has no significant impact on the throughput of various protocols. While AOMDV achieves the best throughput, the DSR achieves the worst value.

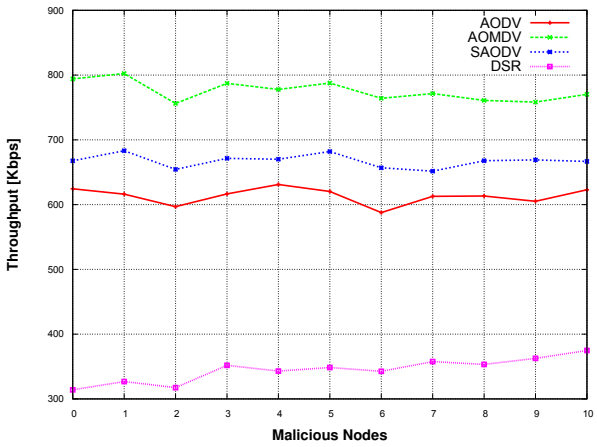

Figure 4.34: Network Throughput under Selfish

Routing overhead of the examined protocols under the selfish attack is shown in Figure 4.35. The figure shows that while both AODV and AOMDV achieve an

approximately constant overhead, the overhead of both SAODV and DSR decreases as a result of this attack. This paradoxical result has been discussed earlier in Section 4.3 and its reason is that the selfish node drops the received routing packets which decreases the overhead. While AOMDV achieves the best overhead, SAODV achieves the worst value under selfish attack.

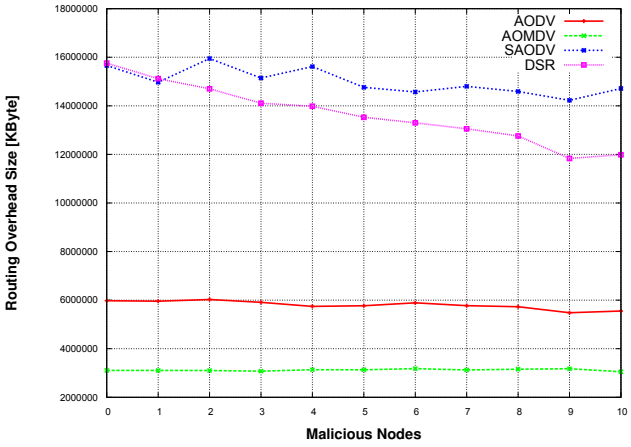

Figure 4.35: Routing Overhead under Selfish

4.8 Summary

In this chapter, we analysed the performance of AODV, DSR, AOMDV and SAODV routing protocols under the blackhole, grayhole, selfish and flooding attacks. We concluded that the blackhole and flooding attacks have dramatic impact on the network performance. The blackhole introduces a fake RREP which affects the network performance and the flooding attack introduces a fake RREQ which affects the network performance as well. As most of the performance metrics depend on the number of received data packets, little change is observed in these metrics under grayhole and selfish attacks because the malicious nodes drop data packets in these attacks. We conclude also that the highest negative impact of malicious nodes in all these different attacks usually appears on static networks and this effect decreases as node mobility increases.

AOMDV is the most resistant protocol to different attacks. While SAODV succeeded in resisting blackhole, grayhole and selfish attacks, it suffers performance

degradation under the flooding attack. SAODV does not forward the routing packets without ensuring authenticity and integrity which explains its success in resisting blackhole, grayhole and selfish attacks. SAODV cannot resist the flooding attack because a malicious node impersonating a non-existent node which could not be discovered by other non-malicious nodes. On the other hand, while SAODV achieves a moderate performance compared to the other protocols, its routing overhead is higher because of the cost of its security features.

Chapter 5

Resisting Flooding Attacks

5.1 Introduction

Reactive MANET routing protocols are vulnerable to a dramatic collapse of network performance under flooding attack. This affects even the secured protocols as discussed in Chapter 4. This chapter introduces two new mechanisms Anti-Flooding (AF) and Flooding Attack Resisting Mechanism (FARM) to resist flooding attacks. These algorithms can be incorporated into any reactive routing protocol. While AF mechanism uses some thresholds and timers to classify nodes as malicious, FARM mechanism uses the concept of Self-Protocol Trustiness (SPT) in which detecting a malicious intruder is accomplished by complying with the normal protocol behaviour and luring the malicious node to give an implicit avowal of its malicious behaviour. Both solutions do not require expensive cryptography or authentication mechanisms or modifications to the packet formats. Using NS2 simulation, we compare the performance of networks under flooding attacks with and without our mechanisms, showing that they significantly reduce the effect of a flooding attack.

The rest of the chapter is organized as follows. Section 5.2 presents the related work. In Section 5.3, Anti-Flooding (AF) mechanism to detect the flooding attack is presented. Section 5.4 introduces the Flooding Attack Resisting Mechanism (FARM) that modifies the AF mechanism. In Section 5.5, a simulation approach and parameters are presented. In Section 5.6, simulation results are given. In Section 5.7, summary is presented.

5.2 Related Work

A number of important algorithms have been introduced to improve MANET routing security, but most of them cannot resist flooding attacks effectively. A malicious node initiating a flooding attack generates a large number of RREQs to non-existent nodes. These RREQ flood out through the MANET and because the destination does not exist, are propagated by all nodes. A node has no way of detecting whether the neighbour that sent the RREQ is malicious or not. All suggested solutions to the flooding attack attempt to classify neighbours as normal or malicious nodes and then suppress malicious ones.

Yi [80] proposed Flooding Attack Prevention (FAP) that defined a neighbour suppression method which prioritizes the node based on the number of RREQ received. A node gets higher priority if it sends fewer RREQ packets. When a malicious node broadcasts large number of RREQ packets, the immediate neighbours of the malicious node observe a high rate of RREQ and then they lower the corresponding priority according to the rate of incoming queries. Forwarding received RREQ depends on the priority value of the sending neighbour. The disadvantage of this algorithm is that it still disseminates flooding packets albeit at a reduced rate.

Peng [81] modified FAP by defining a fixed RREQ threshold. The algorithm assumes that if the number of RREQ packets received from a neighbour exceeds the threshold value, this neighbour is a malicious node and discards all future packets from this malicious node. The disadvantage of this algorithm is obvious if the threshold value is disseminated which introduces an opportunity to a malicious node to subvert the mechanism by sending RREQs under this threshold. Another disadvantage of this algorithm is that it treats a high mobility normal node as if it is a malicious node.

Song [82] defines a Filter-Based (FB) solution that has two threshold values; RATE_LIMIT and BLACKLIST_LIMIT. A RREQ from a neighbour is processed only if the number of previously received RREQ from this neighbour is less than RATE_LIMIT. On the other hand, if the number is greater than BLACK-

LIST_LIMIT, the RREQ is discarded and this neighbour is blacklisted and classified as malicious. If the number of previously received RREQ from this neighbour is greater than RREQ_LIMIT and less than BLACKLIST_LIMIT, the RREQ is queued for processing after a delay expires. A disadvantage of this approach is the ability of the attacker to subvert the algorithm by disseminating thresholds levels and the possibility of permanently suspending a blacklisted neighbour that is not malicious.

Balakrishnan [83] proposed a solution that defines three threshold values; transmission threshold, blacklist threshold and white listing threshold. A RREQ from a neighbour is processed only if the received RREQ rate from this neighbour is less than the transmission threshold; otherwise the node discards the RREQ. If the received RREQ rate from this neighbour is greater than the blacklist threshold, the RREQ is discarded and this neighbour is blacklisted. This algorithm avoids permanently suspending of a blacklisted neighbour by introducing a white listing threshold. A blacklisted neighbour can be returned to normal status if it behaves correctly for a whitelisting time interval.

Venkataraman [84] introduced an algorithm that extends the DSR protocol based on the trust function to mitigate the effects of flooding attack. This algorithm classifies a node neighbours based on a trust value to three categories; friend, acquaintance and stranger. Friend is a trusted node and stranger is a non-trusted node while an acquaintance has the trust value that is greater than a stranger and less than a friend. The algorithm defines a threshold value to each neighbour type. A node decision will be taken based on the neighbour type that sends the RREQ and threshold value of this neighbour type. As a general rule, if a node receives a RREQ from a neighbour, it first checks its relationship class and based on this it checks if this neighbour runs over the relationship class threshold value or not. The node processes the RREQ if this neighbour still running under the relationship class threshold otherwise it discards the RREQ and blacklists this neighbour. The disadvantage of this algorithm is that it cannot support high node mobility. Khartad [85] introduces a modification to this algorithm to extend the algorithm for high

node mobility. A significant disadvantage of this approach is that it depends on a modification of DSR and cannot be adapted to other MANET protocols.

5.3 Anti-Flooding (AF) Mechanism

Anti-Flooding (AF) [86] mechanism is designed to mitigate the effect of the flooding attack on the performance of a MANET routing protocol. The mechanism does not use cryptographic techniques which conserves the power and computation resources. Each node in the network has to monitor the performance of its neighbours to detect if they are attempting to flood the network or not. Malicious nodes will be detected reliably within a very few minutes. The only way for a malicious node to subvert the mechanism is to transmit fake RREQ packets at such a low rate which do not affect the network performance significantly.

The idea is to record for each neighbour the rate at which it transmits RREQs. A node pursuing a flooding attack will be generating a high number of RREQs. If the rate exceeds a threshold, then the neighbour is added to a black list of potential malicious nodes. Once on the black list, RREQs from the black listed node are not forwarded, but they are still recorded. A node can be removed from the black list if its rate of RREQ generation later reduces below the threshold. If the rate continues high, the offending node is queried - only a non-malicious node will respond. After two queries, the neighbour will be suspended for a period, and if its rate is still high after the period has been elapsed it will be declared as malicious. Table 5.1 shows the values of parameters that were used in our simulations. A node implementing the Anti-Flood mechanism behaves as follows:

Table 5.1: AF Mechanism Parameters

RREQ_THRESHOLD	10
RREQ_COUNT_1	7
RREQ_COUNT_2	3
RREQ_TIME_1	5 s
RREQ_TIME_2	2 s
RREP_WAIT_TIME	1 s
TRAFFIC_TIME	10 s
EXCLUDE_TIME	60 s

Algorithm 5.1 AF Neighbour Classification

L: largest number of RREQs received from all neighbours
N: number of RREQ received from a neighbour
T: RREQ threshold
B: neighbour blacklist value (default assigned for normal node)

```
1: calculate L
2: for all neighbour in list do
3:     if N ≥ T then
4:         increment B of neighbour sent L
5:         suspend other neighbours
6:     else
7:         decrement B if not default
8:     end if
9:     reset N
10: end for
```

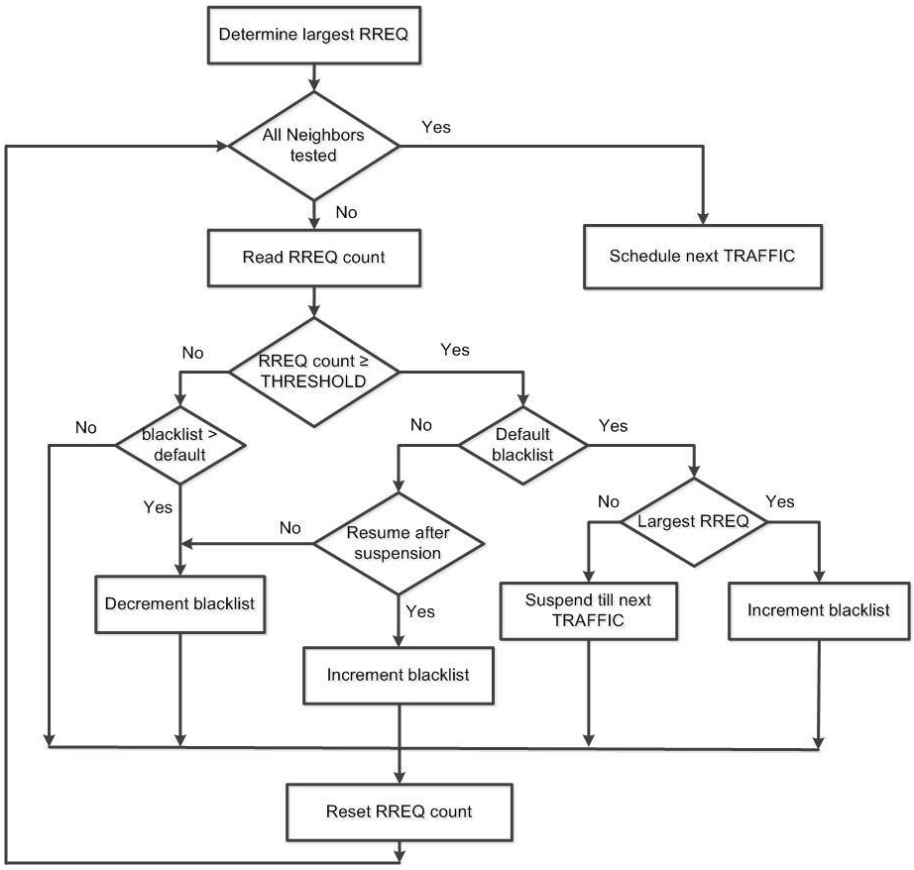

Figure 5.1: AF Neighbour Classification

- Algorithm 5.1 shows that every TRAFFIC_TIME, the number of RREQs received from each neighbour since the last classification update was examined.

- If the number of RREQs received from a neighbour exceeds the threshold RREQ_THRESHOLD, the black_list value of this neighbour is set to 1. If

multiple neighbours exceed the threshold, the black_list value of the neighbour which has sent the largest number of RREQs is set to 1. Other neighbours that exceeded the threshold are suspended for a short period. RREQs from suspended nodes are ignored and not forwarded. Suspension of neighbours except the one with the largest RREQ count allows the mechanism to avoid double counting of RREQs and concentrate on classification of the worst offender. RREQ_THRESHOLD value has been chosen by running AODV on a large number of malicious-free scenarios and observing the largest number of RREQs that can be received in TRAFFIC_TIME as shown in Figure 5.1.

- Algorithm 5.2 and Figure 5.2 show that RREQ packets are processed normally when only received from neighbours with a black_list value of 0. If a RREQ is received from a neighbour with a black_list value of 1, then the node examines how many RREQs have been received in an interval of RREQ_TIME_1. If that is less than RREQ_COUNT_1, the black_list value for that neighbour is reset to 0. If the number exceeds RREQ_COUNT_1, the node tests the authenticity of the neighbour by replying with a fake RREP packet to the RREQ. If the neighbour is malicious, it does not send data to be routed to the destination. If no data is received within RREP_WAIT_TIME, the neighbour's black_list value is set to 2. If the neighbour is not malicious, data is received by the fake RREP originator, which can respond with a RERR so that a new route can be found.

- If a RREQ is received from a neighbour with a black_list value of 2, it re-examines the rate of RREQ received from that neighbour. If the number of RREQ received from this neighbour is less than RREQ_COUNT_2 in an interval of RREQ_TIME_2, it decrements the black_list value to 1. Otherwise the node again sends a fake RREP to the RREQ sender to test its authenticity. If the RREP_WAIT_TIME expires without receiving the data, the node sets the black_list value of this neighbour to 3 and suspends this neighbour for a long period equal to the next TRAFFIC_TIME + EXCLUDE_TIME. This long suspension ensures that if the behaviour of this neighbour has been af-

Algorithm 5.2 AF RREQ Processing

N: number of RREQs received from a neighbour
R: rate of RREQ (N during time period)
L: limit of RREQ rate (depend on B)
B: neighbour blacklist value (default assigned for normal node)

```
1: receive RREQ
2: if neighbour under suspension OR testing then
3:     discard RREQ
4: else
5:     increment N
6:     if B = exclusion then
7:         malicious misbehaviour
8:         discard RREQ
9:     else if B ≠ default then
10:        if R ≥ L then
11:            send fake RREP
12:        else
13:            decrement B
14:        end if
15:        discard RREQ
16:    else
17:        normal RREQ processing
18:    end if
19: end if
```

fected by a malicious node, then that malicious node will have been discovered and excluded during this suspension.

- After the long-time suspension has expired, the node restarts the previous process; it counts again the number of received RREQ from this neighbour and if the number is less than the threshold RREQ_THRESHOLD, it decrements the black_list value to 2. Otherwise it will increment the black_list value to 4.

- If a RREQ is received from a neighbour with a black_list value equal to 4, the node re-monitors the rate of RREQ received from this neighbour. This gives a final chance to the neighbour under suspicion to prove it is innocent and its previous misbehaviour was a result of forwarding malicious' RREQs. If the number of RREQ received from this neighbour is less than RREQ_COUNT_1 in an interval of RREQ_TIME_1, it decrements the black_list value to 3. Otherwise the node sends a fake RREP to the RREQ sender to test its au-

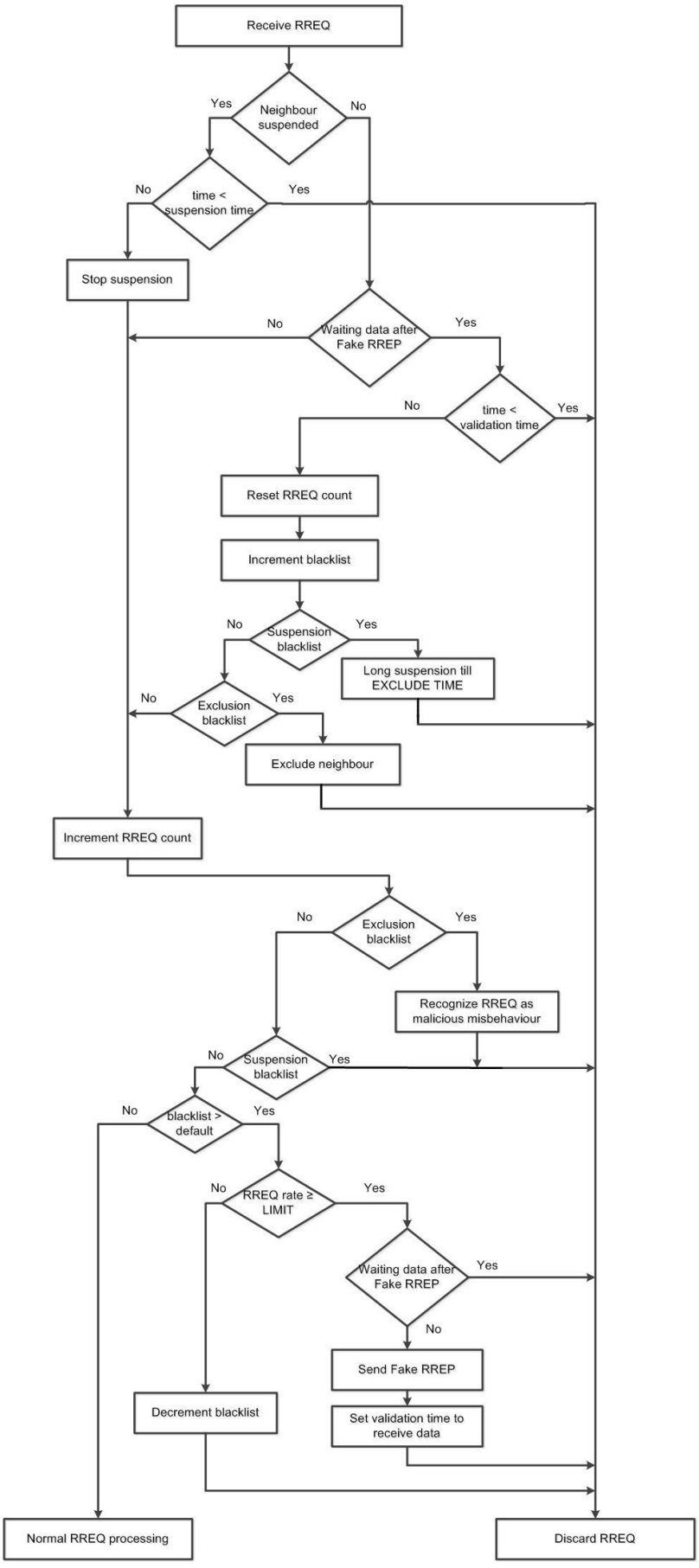

Figure 5.2: AF RREQ Processing

thenticity for the final time. If the RREP_WAIT_TIME expires without receiving the data, the node sets the black_list value of this neighbour to 5, classifies this neighbour as malicious node and removes this neighbour from its routing table. All received RREQ from a neighbour that has a black_list value of 5 will be dropped without processing as a result of its classification as a malicious node.

Although AF mechanism succeeded in discovering malicious nodes within a small time, it has a number of security gaps. Disseminating threshold levels such as RREQ_THRESHOLD, RREQ_COUNT_1 and RREQ_COUNT_2 introduces the ability for an attacker to subvert the algorithm by working below these thresholds. In addition, as RREQs are forwarded through innocent nodes without checking the trustiness of their senders, the algorithm cannot guarantee that the excluded nodes are genuine malicious nodes which increases the number of innocent excluded nodes. The simulation results in the next section clarify that the true exclusion ratio of this algorithm is too small which was the motivation to develop this algorithm by introducing the Flooding Attack Resisting Mechanism (FARM).

5.4 Flooding Attack Resisting Mechanism (FARM)

Flooding Attack Resisting Mechanism (FARM) [87] is designed to mitigate the effect of the flooding attack on the performance of a MANET routing protocol by fast detection of malicious neighbours. FARM modifies AF mechanism to close the security gaps and to overcome its drawbacks such as using fixed thresholds that can be disseminated to malicious nodes and guarantee of genuine malicious nodes exclusion. The mechanism uses the concept of Self-Protocol Trustiness (SPT) [88] that detects a malicious node by following the normal protocol behaviour and luring the malicious node to give an implicit avowal of its malicious behaviour. The mechanism does not use cryptographic techniques which conserves the power and computation resources. Furthermore, the mechanism neither adds new routing packets nor modifies the existing ones.

Each node monitors the behaviour of its neighbours to detect if any of them appears to be attempting a flooding attack. When a node receives more RREQs from a neighbour than a dynamic threshold, it suspects that neighbour and tests it by sending a fake RREP to the next RREQ received from that neighbour. If that neighbour is malicious, it will not have data to send. A non-malicious neighbour will send data, to which the testing node can respond with a RERR packet. Malicious nodes are detected and excluded within a short time and with high reliability compared with existing algorithms. A malicious node cannot subvert the mechanism as there are no fixed thresholds and all variables are randomly set. The algorithm introduces two different variables; trust level and confidence level. The trust level is a node's trust in the network and its assessment of the instantaneous attack threat. Once a node joins a network, it sets its trust level to ***Normal*** mode and updates this trust level to either ***Trust*** or ***Threat*** according to if there is a threat in the network or not. The RREQ threshold is dynamically changed based on the trust level and setting the value is randomly chosen based on the boundaries of these trust levels.

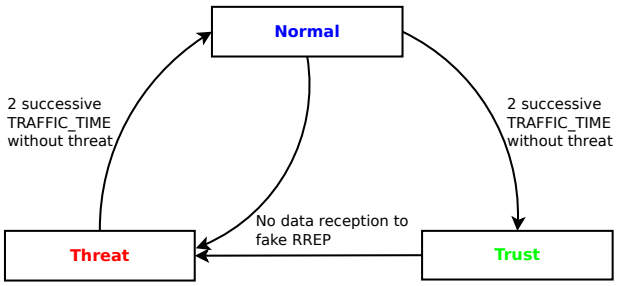

Figure 5.3: FARM Node Trust Level

Figure 5.3 shows the operation of trust levels as a finite state machine. The node assigns a confidence level to each of its neighbours. A confidence level for a neighbour is dynamically changed according to the neighbour behaviour to the testing node fake RREP which consequently changing the testing node trust level. Table 5.2 shows the values of parameters that were used in our simulations. A node implementing the Flooding Attack Resisting Mechanism (FARM) behaves as follows:

Table 5.2: FARM Mechanism Parameters

RREP_WAIT_TIME	1 s
VALIDATE_TIME	2 s
TRAFFIC_TIME	3 s
UPGRADE_LIMIT	2
NEIGHBOUR_LIMIT	2
HOP_LIMIT	5
MAX_CONFIDENCE	11
MIN_THREAT	2
MAX_THREAT	4
MIN_NORMAL	4
MAX_NORMAL	6
MIN_TRUST	6
MAX_TRUST	8

- A node initialises its trust level to ***Normal*** and randomly sets RREQ threshold between MIN_NORMAL and MAX_NORMAL. Once a node sends a fake RREP to test a neighbour and does not receive data within RREP_WAIT_TIME, it changes its trust level to ***Threat*** and randomly sets its threshold between MIN_THREAT and MAX_THREAT.

- The node upgrades it trust level from ***Threat*** to ***Normal*** or from ***Normal*** to ***Trust*** if it does not examine any neighbour during UPGRADE_LIMIT successive TRAFFIC_TIME. A node that set its trust level to ***Trust*** sets its RREQ threshold between MIN_TRUST and MAX_TRUST. The MIN_NORMAL and MAX_NORMAL limits, and their equivalents for ***Threat*** and ***Trust*** levels, are chosen to give a greater range of testing of a neighbour. These three ranges introduce more difficulty for a malicious node looking to subvert our proposed mechanism as it knows neither the trust level nor the random threshold chosen for this level by the victim node as shown in Algorithm 5.3.

- A node initialises the black_list value of all its neighbours to 0 and updates this value based on the neighbour behaviour. RREQ packets are processed normally when received from neighbours with a black_list value of 0. When a node receives a RREQ, it stores two values for each neighbour; the number of RREQs received and the total number of hops received in these RREQs.

Algorithm 5.3 FARM Neighbour Classification

L: largest number of RREQs received from all neighbours
N: number of RREQs received from a neighbour
H: average hop count of RREQs received from a neighbour
T: RREQ dynamic threshold (depend on **Trust** level)
B: neighbour blacklist value (default assigned for normal node)
C: confidence value of a neighbour
M: maximum confidence

```
1: calculate L
2: for all neighbour in list do
3:     calculate H
4:     if N ≥ T then
5:         increment B of neighbour sent L
6:     else
7:         if b≠ default AND H< I then
8:             increment C
9:             if C= M then
10:                reset C
11:                decrement B
12:            end if
13:        end if
14:    end if
15:    reset N
16:    reset H
17: end for
```

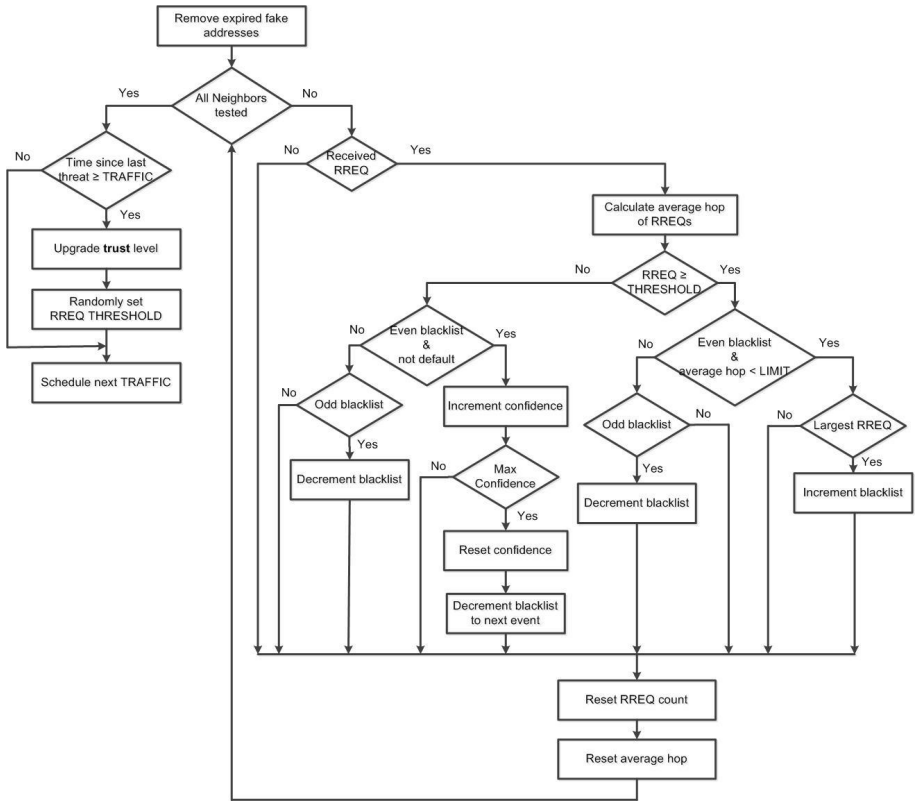

Figure 5.4: FARM Neighbour Classification

- Figure 5.4 show that each TRAFFIC_TIME, A node computes the average hop count for the received RREQs and examines the neighbour that sends the largest number of RREQs during this period. If the largest number of RREQs exceeds the RREQ threshold, which is randomly selected and dynamically changed based on its trust level, and the average hop of all RREQs received from this neighbour is less than HOP_LIMIT, the node sets the black_list value of that neighbour to 1 and sets the time to examine this neighbour to VALIDATE_TIME. If the number of RREQs is less than RREQ threshold, the node increases this neighbour confidence level value until MAX_CONFIDENCE after which the node decreases the black_list value of this neighbour.

- Once a node suspects a neighbour (i.e. black_list = 1), it drops RREQs received from this neighbour that have number of hops greater than NEIGHBOUR_LIMIT which decreases a malicious node's opportunity to flood the network by getting closer to the flooding source. If the node receives a RREQ during VALIDATE_TIME, it sends a fake RREP to this neighbour and stores

Algorithm 5.4 FARM RREQ Processing

N: number of RREQs received from a neighbour
H: hop count of RREQ
L: hop count limit (small to filter RREQs)
B: neighbour blacklist value (default assigned for normal node)
T: RREQ dynamic threshold (depend on **Trust** level)

```
1: receive RREQ
2: increment N
3: Add H
4: if B = exclusion then
5:     malicious misbehaviour
6:     discard RREQ
7: else if B ≠ default AND H ≥ L then
8:     if neighbour under testing then
9:         switch to Threat mode
10:        randomly set T
11:        increment B
12:    else
13:        send fake RREP
14:        store fake Src & Dst addresses
15:    end if
16:    discard RREQ
17: else
18:    normal RREQ processing
19: end if
```

the source and destination addresses of this RREQ in a trustiness table for later examination. The node also sets an expiry time for this entry to avoid the table inflation.

- Algorithm 5.4 clarifies that if the node later receives data from that suspected neighbour, it is assumed that this neighbour is not a malicious node and resets this neighbour's black_list value. If it does not receive data within RREP_WAIT_TIME; it sets the black_list value of this neighbour to 2.

- The algorithm is repeated three times to ensure that the non-reception of data after examining a neighbour is not a result of link failure or network congestion. If the black_list value of a neighbour reaches 6, the node classifies that neighbour as a malicious node and removes it from its routing table and drops any upcoming RREQs received from this neighbour without processing as shown in Figure 5.5.

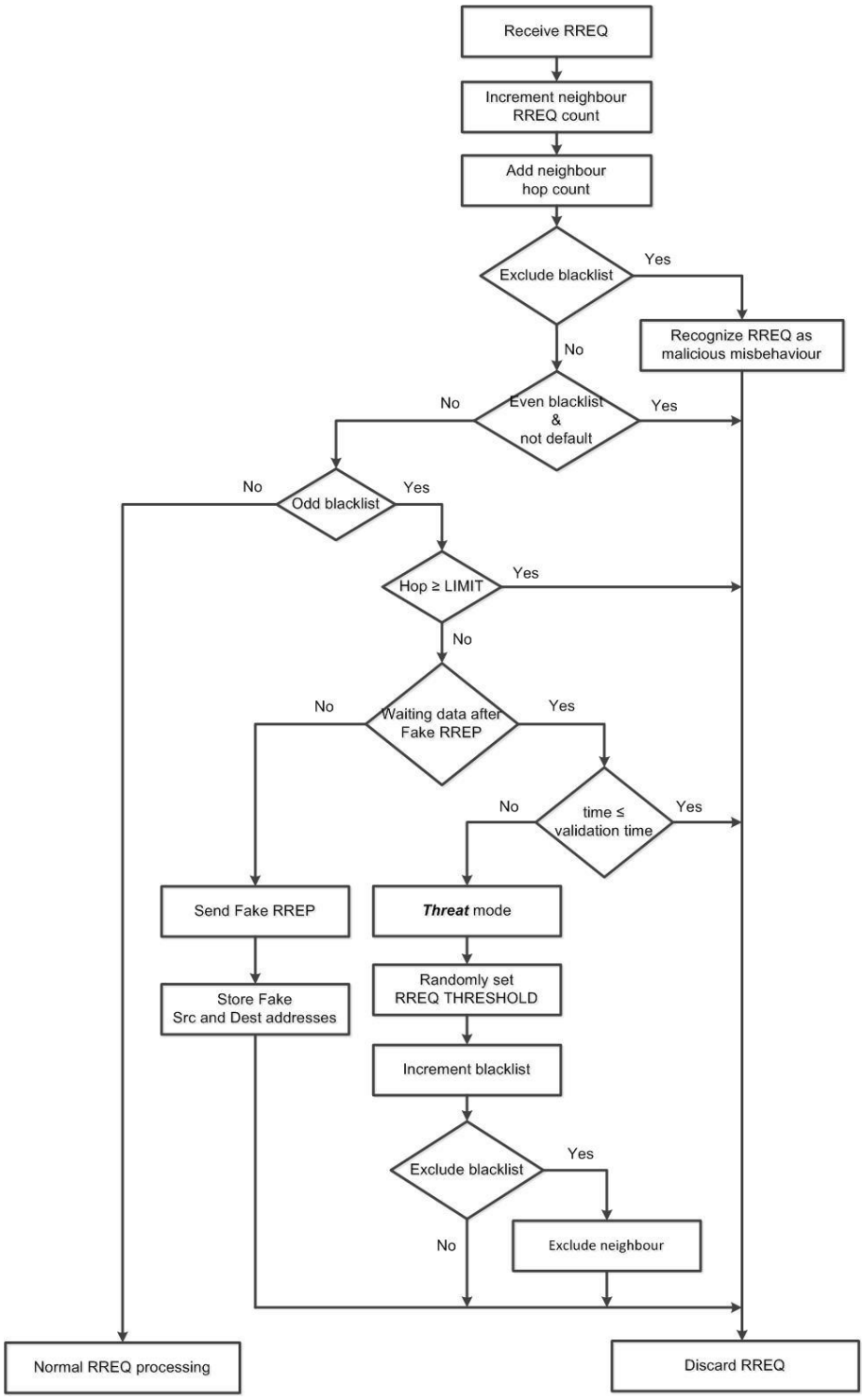

Figure 5.5: FARM RREQ Processing

FARM algorithm includes strong features that do not allow malicious nodes to subvert it. The algorithm has no fixed thresholds; that could be disseminated to malicious nodes and allow these malicious nodes to work under these thresholds. Instead of this, the algorithm sets RREQ threshold randomly depending on the trust level of a node. The adjacent threshold limits of the three level of trustiness

introduce a difficulty for a malicious node to subvert the algorithm. As shown from Table 5.2, the RREQ threshold is randomly set between 2 and 8 which is a big range that makes it difficult for a malicious node to estimate the threshold value.

In addition, FARM takes into account network problems such as link failure and congestion by giving a neighbour 2 consecutive chances to prove its normal behaviour. After the third consecutive misbehaviour by receiving a RREQ and did not receive data later in response to the fake RREP, a node will be sure that this neighbour attempts to flood the network. Moreover, this represents as well a warning for a malicious node that its bad behaviour has been discovered asking it to give up on this. Our proposed algorithm does not care about all nodes in the MANET; each node cares only about its neighbours (1 hop only from it). If a node sends a fake RREP to a neighbour and it does not later receive data from this neighbour regarding to this RREP; it will be sure that this neighbour is a malicious or a colluding node and it should be excluded after multiple examination. If each node succeeded in excluding its malicious neighbours, this would guarantee malicious-free routes. To ensure that the excluded node is a malicious node, our algorithm ignores RREQs from a neighbour with a non-zero black_list value and the number of hops in the RREQ greater than NEIGHBOUR_LIMIT.

Moreover, a node can temporarily increment the black_list value of an innocent neighbour as a result of forwarding malicious RREQs. Later, when this innocent neighbour succeeded in discovering and excluding the malicious node, a change of the behaviour of this innocent neighbour is observed by the node. We suggest that a node has to increment this innocent neighbour confidence level until MAX_CONFIDENCE after that it decrements its black_list value to avoid a permanent effect of malicious node on other innocent nodes.

As we will see later in simulations, FARM algorithm succeeded in detecting and excluding most of malicious neighbours in a short time. A malicious node is detected whenever it does not send data after it is examined by a fake RREP. FARM algorithm does not assume that the attacker has to continue its malicious behaviour; but it guarantees that whenever a malicious node starts its bad behaviour, it will

be discovered within a short time. So, a malicious node has two choices; either to continue its bad behaviour by sending RREQs and expose itself for detection and exclusion or it sends RREQs under the lowest threshold MIN_THREAT (i.e. 1 RREQ) every TRAFFIC_TIME which is a very low rate even compared to the normal RREQ rate to be safe from detection which leads to the same result of preventing the flooding attack.

5.5 Simulation Approach

NS-2 simulator [69] is used to simulate flooding attack. The simulation is used to analyse the performance of the networks under the flooding attacks with and without our new two mechanisms AF and FARM. The parameters used are shown in Table 5.3. While we examined our proposed mechanisms on both UDP and TCP traffic and the mechanisms succeeded in detecting flooding neighbours and enhancing the network performance for both traffic, the chapter is focused on the results of the proposed mechanisms on the TCP traffic only. We examined our proposed mechanisms for different number of nodes (25, 50, 75 and 100) and different node speeds (0, 5, 10, 15, 20, 25 and 30 m/s). Similarly, only the case of 100 node networks is reported, corresponding to a high density of nodes. This gives malicious nodes a high number of neighbours. We tested our algorithms on different simulation areas (500 m^2 and 1000 m^2) to ensure the effect of nodes' density on our algorithm. Although, we report the higher area in this chapter for the consistency of the thesis, results for the smaller area simulation are presented in [86]. We choose a large simulation time to ensure that the vast majority of malicious nodes have been detected especially for scenarios with a large number of malicious nodes. The highest negative impact of malicious nodes usually appears on static networks and this effect decreases as node mobility increases [77], so we report here the case of static networks.

Our flooding attack model assumes that a malicious node periodically generates a RREQ packet from a non-existent source to a non-existent destination each random interval between MIN_FLOOD and MAX_FLOOD. The attacker constructs a fake

Table 5.3: Resisting Flooding Attacks Simulation Parameters

Simulation Time	600 s
Simulation Area	1000 m^2
Number of Nodes	100
Number of Connections	150
Number of Malicious Nodes	0 - 10
Node Speed	0 - 30 m/s
Pause Time	10 s
Traffic Type	TCP
MIN_FLOOD	0.25 s
MAX_FLOOD	0.50 s

RREQ that includes a randomly generated hop count between 2 and 4. The value 2 ensures that a malicious node spoofs its neighbours that it forwards a RREQ received from another node, while the value 4 ensures that this RREQ travels at least (NETWORK_DIAMETER - 4) hops to flood the network. On the other hand, the generated fake RREP which is used to test the trustiness of a neighbour is unicast to this neighbour with a number of hops randomly chosen between 2 and 4 to spoof this neighbour that it has the best route (i.e. 1 to 3 hop counts only from the RREQ source). The RREP originator is also chosen randomly and the destination sequence number value of this RREP is set to the received corresponding one in the RREQ plus a randomly generated number between 10 and 30 to spoof this neighbour about the freshness of this RREP.

5.6 Simulation Results

AF and FARM algorithms achieve different levels of success in excluding flooding nodes. As the AF mechanism achieves a smaller success than the FARM mechanism, we include only the FARM effect on the network performance in this chapter. FARM succeeded in detecting and excluding more than 80% of malicious neighbours in the simulation time with a highly trusted ratio that exceeds 90% in AODV, DSR and SAODV. Although FARM achieves a smaller success in AOMDV, it can be adapted by modifying the trust levels boundary values. Trust levels boundaries are chosen based on calculating the average number of RREQs disseminated through

the network each TRAFFIC_TIME if the network does not have malicious nodes. The boundaries stated in Table 5.2 are used for AODV, DSR and SAODV protocols. On the other hand, as the rate of RREQs traversing the network in a period for AOMDV are usually more than double the values of the other protocols, we use the same boundary values stated in Table 5.2 for AOMDV while we reduce the TRAFFIC_TIME to half of its value in other protocols (i.e. TRAFFIC_TIME = 1.5 sec). Details of the simulations are presented in the following sections.

5.6.1 Resisting Flooding Attacks in AODV

In this section, we evaluate the performance of AODV under flooding attacks with and without our two mechanisms and compare with the results of the Filter-Based (FB) [82] algorithm. Our simulation results show that FARM-AODV achieves a higher exclusion ratio of genuine malicious nodes than other algorithms.

Figure 5.6 shows that FARM-AODV ensures that if a neighbour is excluded, in 98% of cases this neighbour is a malicious node. On the other hand, FB-AODV and AF-AODV have very low true exclusion ratio.

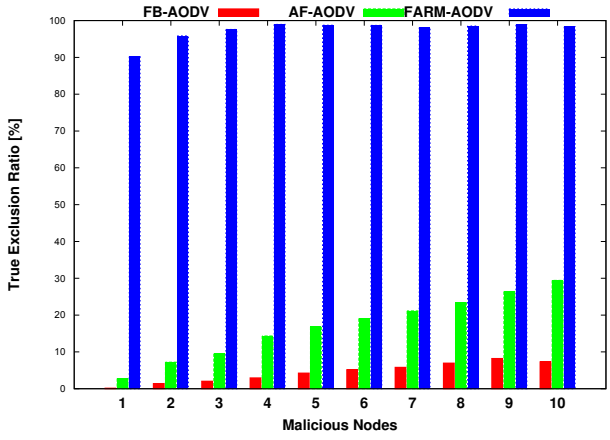

Figure 5.6: AODV True Exclusion Ratio

The effect illustrated by the true exclusion ratio is more noticeable if we combine it with the total number of neighbours excluded during the simulation which is shown in Figure 5.7. The figure shows the effect of dramatic true exclusion ratio of both FB-AODV and AF-AODV on excluding high number of victim nodes. The total number

of neighbours that is wrongly excluded during the simulation is too high for both FB-AODV and AF-AODV. Although FARM-AODV excludes the lowest number of neighbours, it guarantees that the vast majority of these excluded neighbours are genuine malicious nodes.

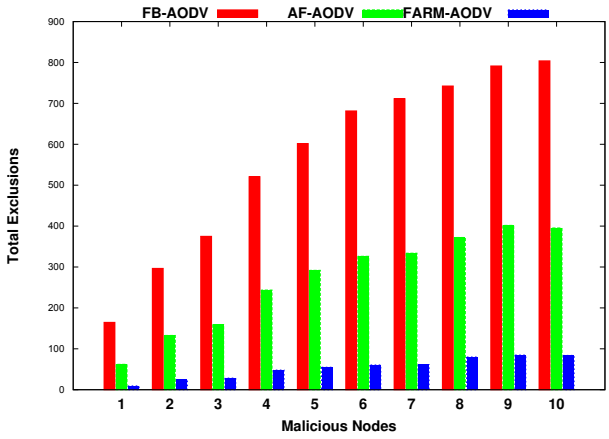

Figure 5.7: AODV Total Exclusions

Our simulation shows that regardless of the number of nodes and the number of malicious nodes in the network, a node will detect a malicious neighbour within a short time. Figure 5.8 shows the proportion of malicious nodes that have been detected as time progresses. For clarity, we only show the results for 1, 4, 7 and 10 malicious nodes. As the simulation time increases, FARM succeeded in detecting and excluding malicious nodes up to 75% of malicious neighbours within 600 seconds. The mechanism succeeded in excluding high ratio of the flooding neighbours after 120 seconds from the beginning of the simulation because most of the genuine RREQs and RREPs are sent during this period. The mechanism continues to exclude more malicious neighbours after that at a low rate as a result of small number of RREQs.

The effect of FARM algorithm on the AODV packet delivery ratio is shown in Figure 5.9. While the flooding attack shows a slight degradation on the PDR of AODV especially for large number of malicious nodes, FARM-AODV achieves an approximately constant PDR regardless of the number of malicious nodes.

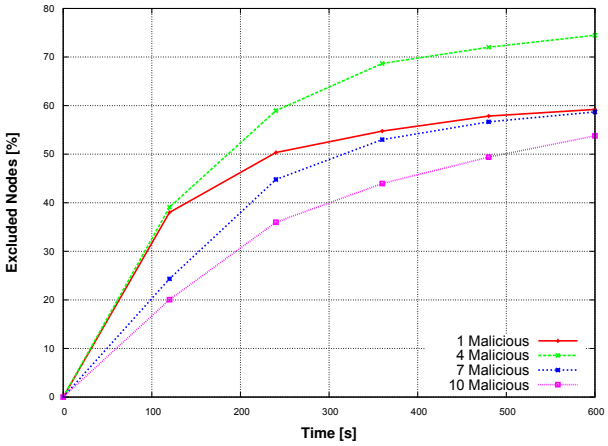

Figure 5.8: FARM-AODV Malicious Discovery Ratio

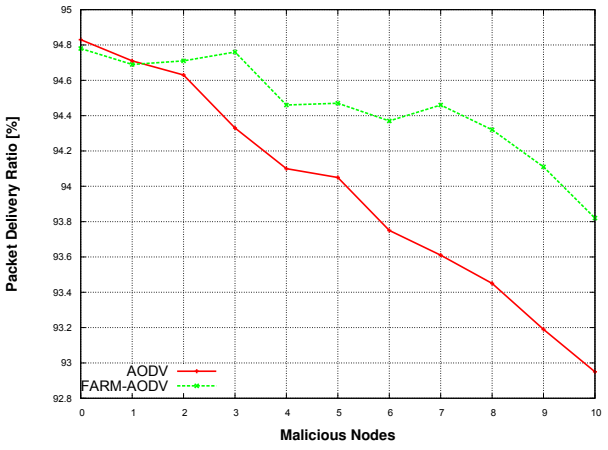

Figure 5.9: FARM Impact on AODV Packet Delivery Ratio

Figure 5.10 shows the effect of FARM algorithm on the AODV network throughput. From the previous discussions in Chapter 4 and from this figure, we notice that the flooding attack has a severe negative impact on the network throughput. FARM improves AODV network throughput by approximately 3% for each malicious node and the enhancement becomes huge for large number of malicious nodes. While the throughput of AODV dramatically decreases as the number of malicious nodes increases, FARM-AODV achieves a slow rate degradation of throughput as a result of continuous detection and exclusion of malicious neighbours.

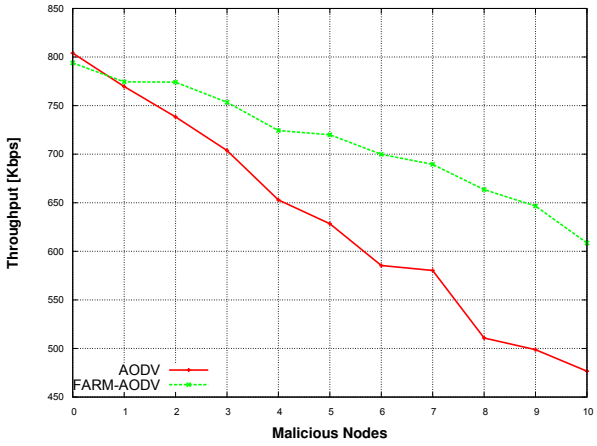

Figure 5.10: FARM Impact on AODV Network Throughput

The effect of FARM algorithm on the AODV delay is shown in Figure 5.11. While the delay increases as the number of malicious nodes increases in AODV, FARM-AODV achieves better average delay.

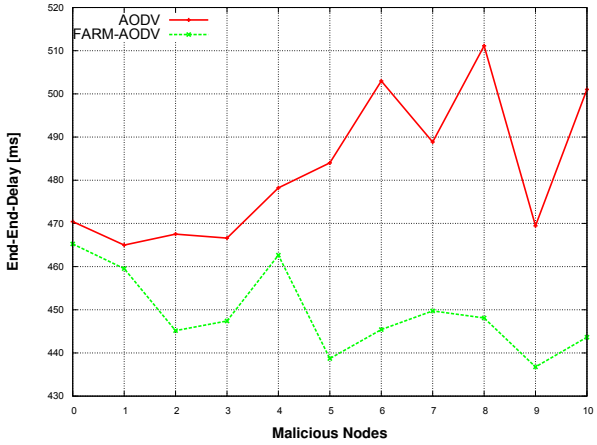

Figure 5.11: FARM Impact on AODV End-to-End Delay

Figure 5.12 shows the effect of FARM algorithm on the AODV normalized routing load. The result shows that while the normalized routing load of AODV increases as the number of malicious nodes increases especially for large number of malicious nodes, it has a less significant change for FARM-AODV.

Figure 5.13 shows the effect of FARM algorithm on the AODV routing overhead. FARM succeeded in reducing overhead of AODV by approximately 3% for each malicious node.

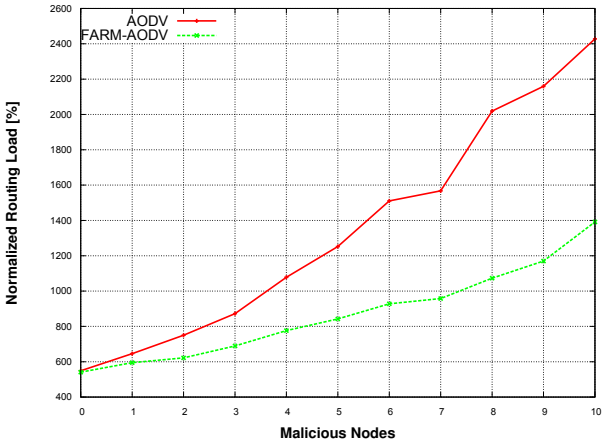

Figure 5.12: FARM Impact on AODV Normalized Routing Load

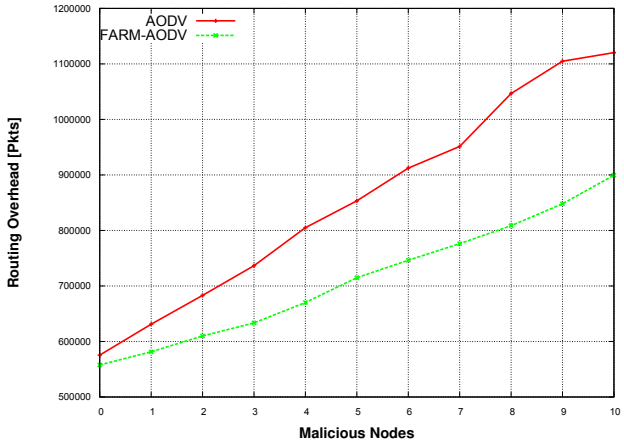

Figure 5.13: FARM Impact on AODV Routing Overhead

Figure 5.14 shows the effect of FARM algorithm on the AODV routing discovery latency. The result shows that while RDL of AODV increases dramatically as the number of malicious nodes increases, FARM achieves a huge improvement in RDL especially for large number of malicious node.

5.6.2 Resisting Flooding Attacks in DSR

In this section, we compare the performance of DSR under flooding attacks with and without our two mechanisms. Our simulation results show that FARM mechanism achieves a higher exclusion ratio of genuine malicious nodes than AF mechanism.

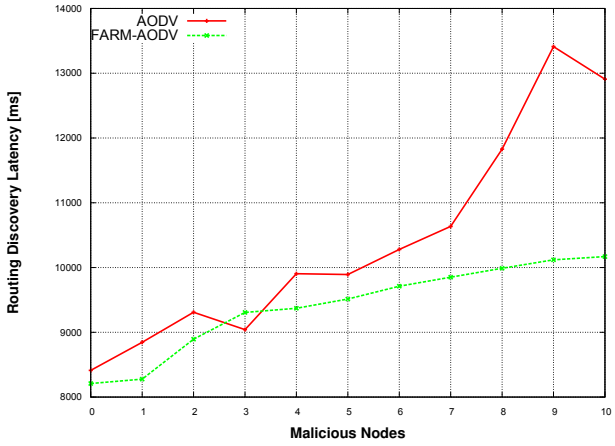

Figure 5.14: FARM Impact on AODV Route Discovery Latency

Figure 5.15 shows that our algorithm ensures that if a neighbour is excluded, in 95% of cases this neighbour is a malicious node which is slightly smaller than AODV ratio as shown in Figure 5.6.

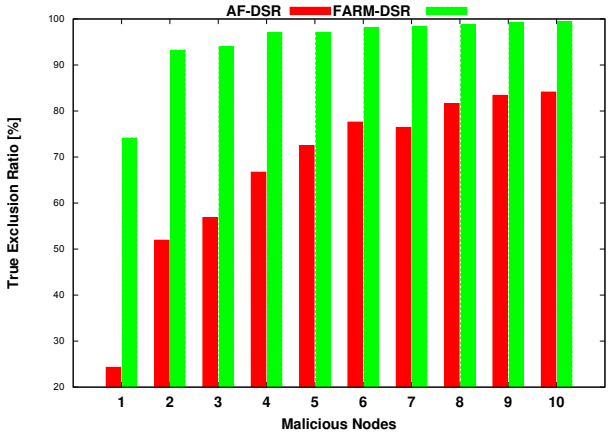

Figure 5.15: DSR True Exclusion Ratio

The effect illustrated by true exclusion ratio is more remarkable if we combine it with the total number of neighbours excluded during the simulation which is shown in Figure 5.16. While FARM has a smaller number of exclusions than AF, the vast majority of them are malicious nodes. Figure 5.7 and Figure 5.16 show that FARM achieves as well higher number of exclusions in DSR than AODV.

Figure 5.17 shows the proportion of malicious nodes that have been detected as time progresses. The figure shows that as the simulation time increases the

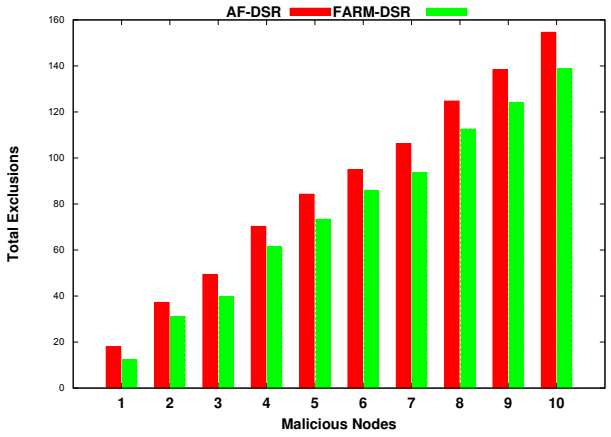

Figure 5.16: DSR Total Exclusions

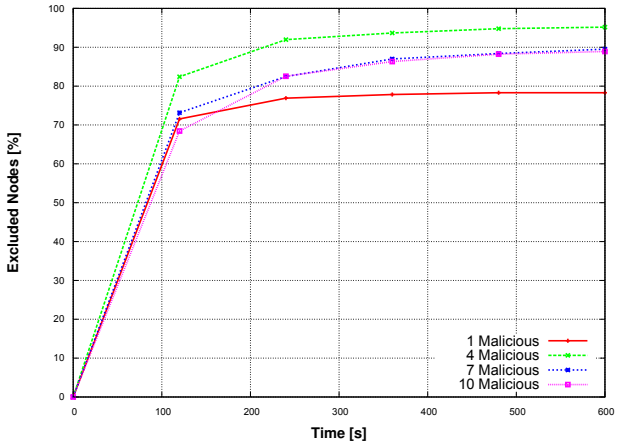

Figure 5.17: FARM-DSR Malicious Discovery Ratio

mechanism succeeded in detecting and excluding malicious nodes up to 95% of malicious neighbours within 600 seconds which is higher than its ratio in AODV as shown in Figure 5.8.

As a result of excluding most of the malicious neighbours, FARM achieves approximately constant performance in all metrics regardless of the number of malicious nodes joining the network. This improvement has a positive impact in the absence of malicious nodes as well. This is because a node does not forward all received RREQs and it drops RREQs received from suspected nodes which decreases the overhead and frees the medium to deliver other data. The following figures present some of these enhancements. The effect of FARM algorithm on

the DSR packet delivery ratio is shown in Figure 5.18. While the flooding attack has a dramatic negative impact on the PDR of DSR, FARM-DSR achieves an approximately constant PDF regardless of the number of malicious nodes.

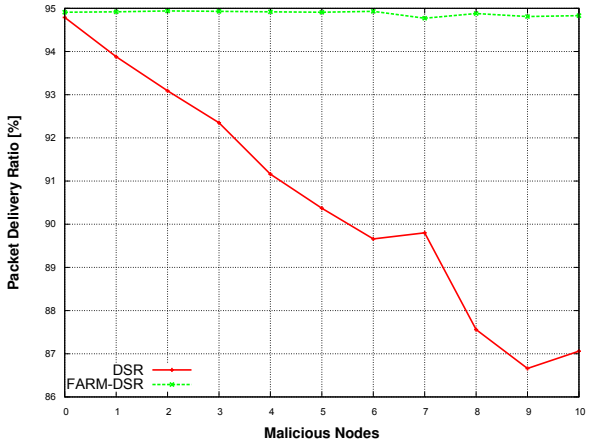

Figure 5.18: FARM Impact on DSR Packet Delivery Ratio

Figure 5.19 shows the effect of FARM algorithm on the DSR network throughput. FARM-DSR achieves a constant throughput compared to original DSR which has dramatic degradation as the number of malicious nodes increases.

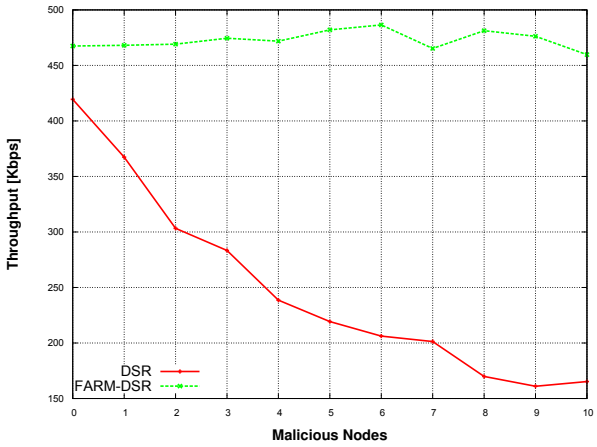

Figure 5.19: FARM Impact on DSR Network Throughput

The effect of FARM algorithm on the DSR end-end-delay is shown in Figure 5.20. While the delay increases as the number of malicious nodes increases in DSR, FARM-DSR achieves approximately constant delay.

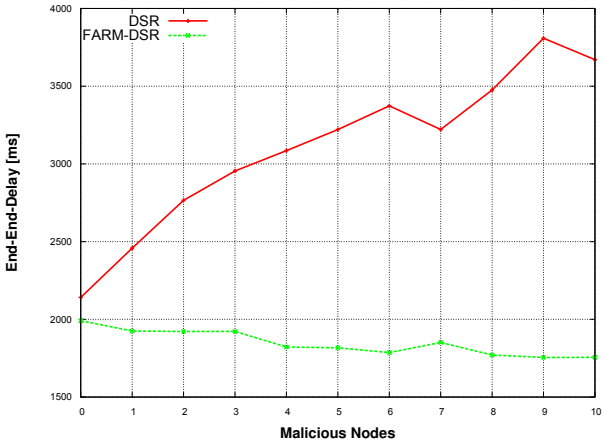

Figure 5.20: FARM Impact on DSR End-to-End Delay

Figure 5.21 shows the effect of FARM algorithm on the DSR routing overhead. The result shows that FARM-DSR achieves a constant overhead compared to the original DSR. One of the most interesting results is the enhancement of overhead even in the absence of malicious nodes. This is, as mentioned earlier, because of the selective droppings feature of RREQs by a node.

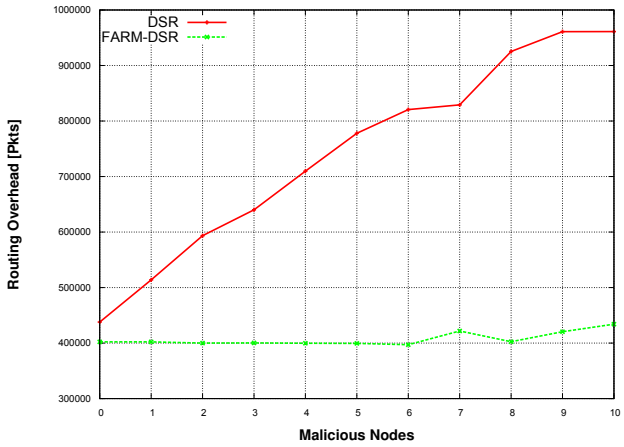

Figure 5.21: FARM Impact on DSR Routing Overhead

5.6.3 Resisting Flooding Attacks in SAODV

In this section, we compare the performance of SAODV under flooding attacks with and without our two mechanisms. Our simulation results show that FARM mechanism achieves a higher exclusion ratio of genuine malicious nodes than AF mechanism. Figure 5.22 shows that FARM ensures that if a neighbour is excluded,

in 90% of cases this neighbour is a malicious node especially for large number of malicious nodes. This is slightly less than its value for both AODV and DSR if we compare this figure to Figure 5.6 and Figure 5.15.

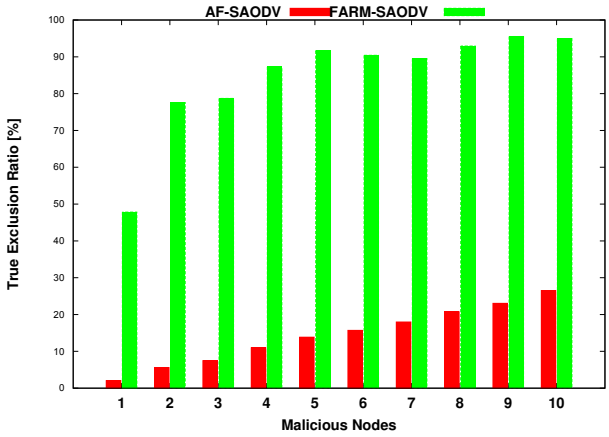

Figure 5.22: SAODV True Exclusion Ratio

As mentioned earlier, the effect illustrated by true exclusion ratio is more remarkable if we combine it with the total number of neighbours excluded during the simulation which is shown in Figure 5.23. The figure shows that while FARM has a smaller number of exclusions than AF, the vast majority of them are malicious nodes. FARM achieves a smaller number of exclusions in SAODV than either AODV or DSR.

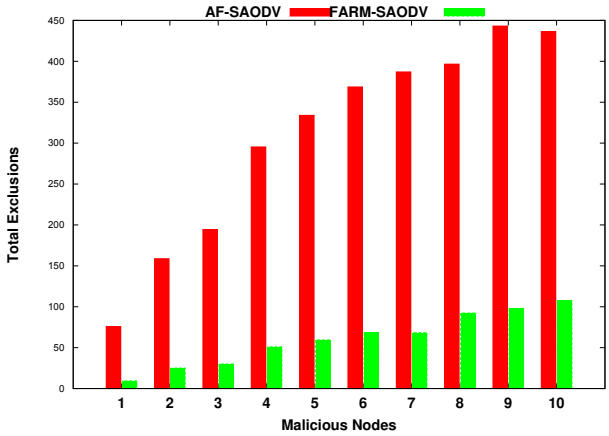

Figure 5.23: SAODV Total Exclusions

Figure 5.24 shows the proportion of malicious nodes that have been detected as time progresses. The figure shows that as the simulation time increases the

mechanism succeeded in detecting and excluding malicious nodes up to 80% of malicious neighbours within 600 seconds which is higher ratio than AODV and less than DSR.

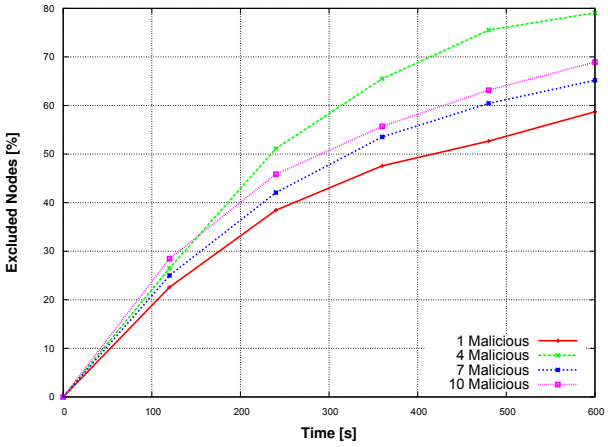

Figure 5.24: FARM-SAODV Malicious Discovery Ratio

FARM achieves approximately a small difference in all performance metrics regardless of the number of malicious nodes. The following figures prove the validity of this advantage. Figure 5.25 shows the effect of FARM algorithm on the SAODV network throughput. While the throughput of FARM-SAODV slightly decreases compared to the original SAODV, the variations of its values regardless of the number of malicious nodes is smaller than SAODV which has a remarkable change.

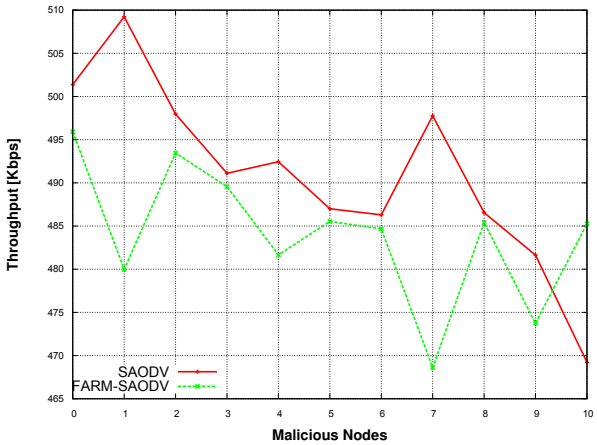

Figure 5.25: FARM Impact on SAODV Network Throughput

Figure 5.26 shows the effect of FARM algorithm on the SAODV routing overhead. FARM-SAODV achieves a better routing overhead and the variations of its values

regardless of the number of malicious nodes are smaller than SAODV which has a bigger change.

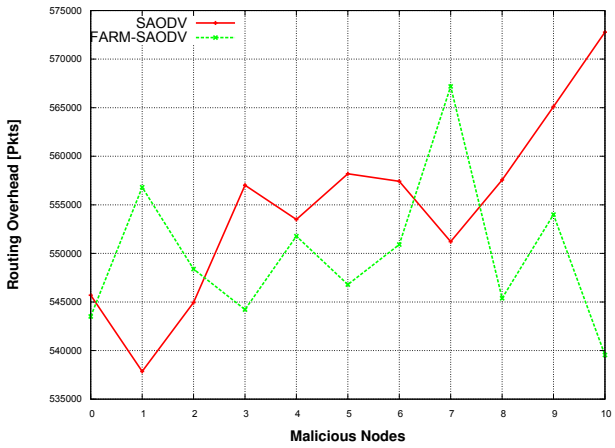

Figure 5.26: FARM Impact on SAODV Routing Overhead

5.6.4 Resisting Flooding Attacks in AOMDV

In this section, we compare the performance of AOMDV under flooding attacks with and without our two mechanisms. Our simulation results show that while FARM mechanism achieves a small exclusion ratio of genuine malicious nodes, its true exclusion ratio is better than AF mechanism. FARM excludes smaller number of nodes with a higher ratio of exclusion.

Figure 5.27 shows that FARM ensures that if a neighbour is excluded, in 75% of cases this neighbour is a malicious node. Although this is a small trustiness value, it can be increased by modifying the boundaries values of the trust levels. This smaller ratio compared to all the other protocols is achieved because the number of RREQs traversing the network in a period for AOMDV is usually more than double the number for the other protocols. Finding minimum/maximum trust level boundary values will produce higher exclusion ratio.

As discussed earlier, the effect illustrated by the true exclusion ratio is more remarkable if we combine it with the total number of neighbours excluded during

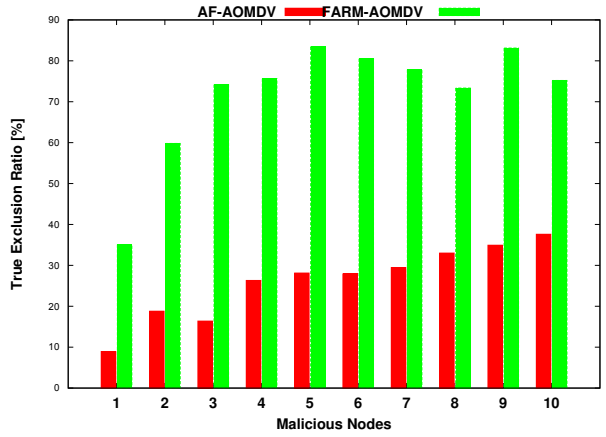

Figure 5.27: AOMDV True Exclusion Ratio

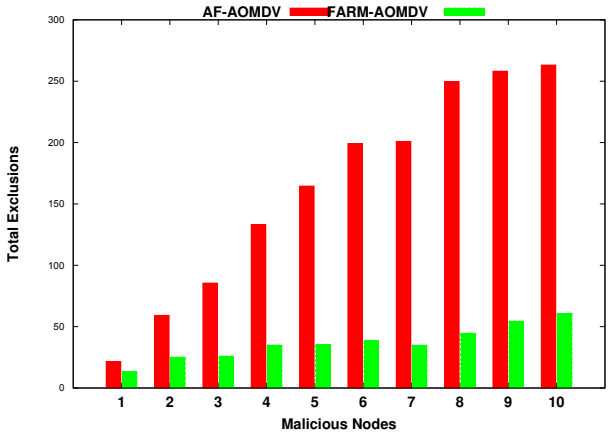

Figure 5.28: AOMDV Total Exclusions

the simulation which is shown in Figure 5.28. The figure shows that FARM has a smaller number of exclusions than AF.

Figure 5.29 shows the proportion of malicious nodes that have been detected as time progresses. The figure shows that as the simulation time increases the mechanism succeeded in detecting and excluding malicious nodes up to more than 80% of malicious neighbours within 600 seconds and this ratio is high for a large number of malicious nodes.

As a result of low exclusion ratio, FARM achieves a smaller improvement in the network performance than the original AOMDV and this improvement is more re-

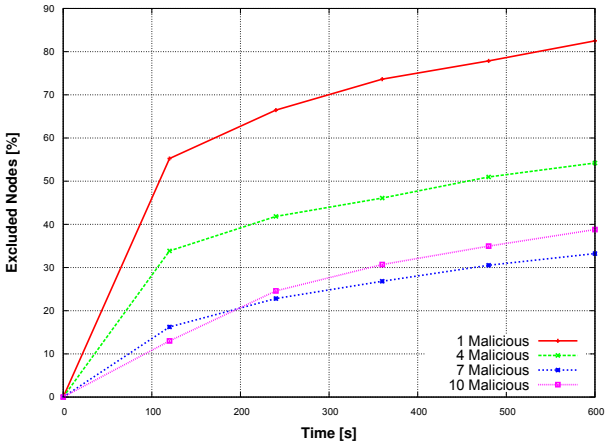

Figure 5.29: FARM-AOMDV Malicious Discovery Ratio

markable especially with a large number of malicious nodes. The following figures present some of these improvements. Figure 5.30 shows the effect of FARM algorithm on the AOMDV network throughput. While the throughput of AOMDV with and without FARM decreases as the number of malicious nodes increases, FARM achieves an improvement up to 13% of the original AOMDV when the number of malicious nodes is 10.

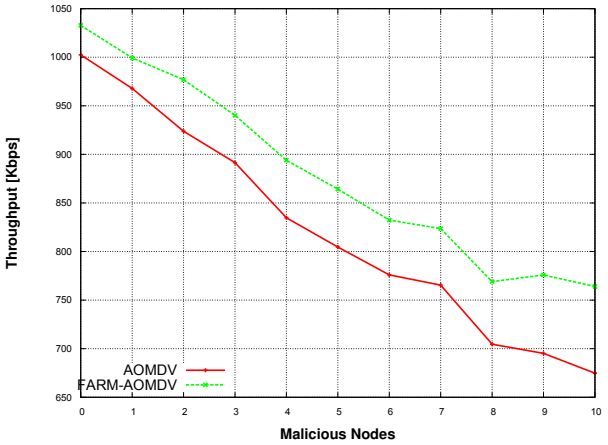

Figure 5.30: FARM Impact on AOMDV Network Throughput

Figure 5.31 shows the effect of FARM algorithm on the AOMDV end-end-delay. The result shows that FARM-AOMDV achieves a better delay than the original AOMDV which increases dramatically as the number of malicious nodes increases.

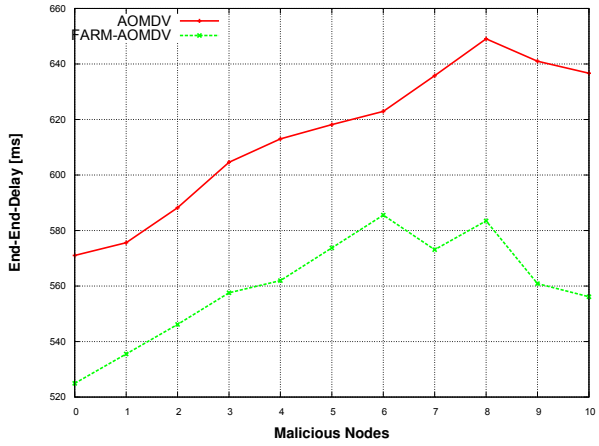

Figure 5.31: FARM Impact on AOMDV End-to-End Delay

Figure 5.32 shows the effect of FARM algorithm on the AOMDV routing overhead. While the overhead of AOMDV with and without FARM increases as the number of malicious nodes increases, FARM achieves an improvement up to 12% of the original AOMDV when the number of malicious nodes is 10.

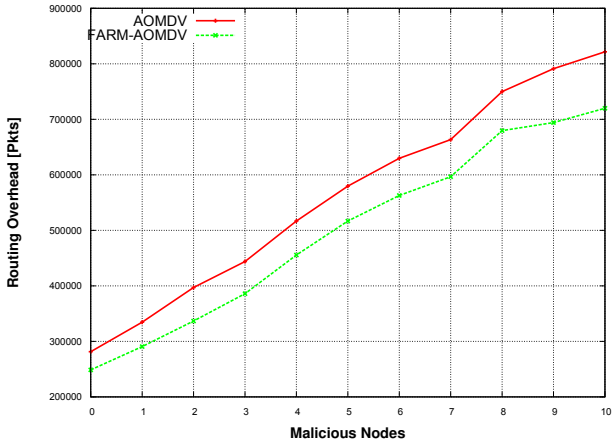

Figure 5.32: FARM Impact on AOMDV Routing Overhead

5.7 Summary

A flooding attack has a dramatic impact on the MANET routing protocols performance even for secured protocols such as SAODV. We proposed two new mechanisms Anti-Flooding (AF) and Flooding Attack Resisting Mechanism (FARM) to resist flooding attacks that can be incorporated into any reactive routing protocol.

While AF mechanism uses some thresholds and timers to classify nodes as malicious, FARM mechanism uses the concept of Self-Protocol Trustiness (SPT) in which detecting a malicious intruder is accomplished by complying with the normal protocol behaviour and luring the malicious node to give an implicit avowal of its malicious behaviour. FARM mechanism is designed to close the security gaps and overcome the drawbacks of AF mechanism such as fixed thresholds that enable malicious nodes to subvert it. It succeeded in detecting and excluding a high ratio of flooding nodes in a short time. Neither solution requires expensive cryptography or authentication mechanisms or modifications to the packet formats.

Using NS2 simulation, we compare the performance of networks under flooding attacks with and without our mechanisms, showing that it significantly reduces the effect of a flooding attack. Both AF and FARM algorithms achieve success in excluding flooding nodes with different ratios. When incorporated into AODV, DSR or SAODV, FARM succeeded in detecting and excluding more than 80% of malicious neighbours in the simulation time with a highly trusted ratio exceeds than 90%. Used with AOMDV, FARM was less successful but it could be improved by modifying the trust level boundary values.

Chapter 6

Resisting Blackhole Attacks

6.1 Introduction

Reactive MANET routing protocols are vulnerable to a dramatic collapse of network performance in the presence of blackhole attack. The chapter introduces two new mechanisms Anti-Blackhole (AB) and Blackhole Resisting Mechanism (BRM) to resist such attacks that can be incorporated into any reactive routing protocol. Both AB and BRM mechanisms use the concept of Self-Protocol Trustiness (SPT) in which detecting a malicious intruder is accomplished by complying with the normal protocol behaviour and luring the malicious node to give an implicit avowal of its malicious behaviour. Neither solution requires expensive cryptography or authentication mechanisms or modifications to the packet formats. Using NS2 simulation, we compare the performance of networks under blackhole attacks with and without our mechanisms, showing that it significantly reduces the effect of a blackhole attack.

The rest of the chapter is organized as follows. Section 6.2 presents the related work. In Section 6.3, the Anti-Blackhole (AB) mechanism to detect the blackhole attack is presented. Section 6.4 introduces the Blackhole Resisting Mechanism (BRM) that modifies the AB mechanism. In Section 6.5, a simulation approach and parameters are presented. In Section 6.6, simulation results are given. Section 6.7 summarizes the findings.

6.2 Related Work

Since on-demand routing protocols have been introduced, many significant algorithms have been proposed to secure MANETs against blackhole attacks. Some of these solutions are designed based on using cryptographic techniques to secure the routing packets. Although these solutions introduce high immunity to the blackhole attack, network nodes suffer from the high computations required which does not suit the characteristics of MANET. Other solutions suggest modification to the routing protocols by adding some packets, modifying the existing packets or changing the procedure of these protocols. While a small number of these solutions are introduced as a mechanism that can be incorporated to any routing protocols, the majority of these solutions are designed for a specific routing protocol to detect and defend against this type of attack.

Such solutions focus their suggested mechanisms on two characteristics of the RREP received from a blackhole node; the first is that this reply is usually received before any other replies as a result of blackhole node sending this fake reply without checking its route table. The second is that this fake RREP usually contains a much higher sequence number relative to the RREQ because the blackhole node tries to convince its neighbours it has a fresh route to the destination node. Suggested solutions classify the attack as single blackhole attack or cooperative blackhole attack. In a cooperative attack, multiple malicious nodes attempt to subvert the routing protocol. Some solutions detect single attacks and give acceptable protection. Others attempt to detect cooperative attacks. All these solutions make assumptions about blackhole behaviour and cannot guarantee that the excluded nodes are genuine blackhole nodes. Forwarding a RREP received from a blackhole can result in a node being wrongly classified as malicious. In this section we introduce some of the existing algorithms used to avoid the blackhole attack.

SAODV [54] is an enhancement of the AODV routing protocol. The protocol operates mainly by appending an extension message to each AODV message. The extension messages include a digital signature of the routing packet using the private

key of the original sender of the routing message and a hash value of the hop count. SAODV uses asymmetric cryptography to authenticate all non-mutable fields of routing messages as well as hash chain to authenticate the hop count (the only mutable) field. Since all fields except the hop count of routing messages are non-mutable they can be authenticated by verifying the signature using the public key of the message originator. So, when a routing message is received by a node, the node verifies the signature of the received packet. If the signature is verified, the node computes the hash value of the hop count, if the routing message is RREQ or RREP, and compares it with the corresponding value in the SAODV extension. If they match, the routing message is valid and will be forwarded with an incremented hop count and a new hash value. If the destination receives the RREQ, it verifies the RREQ signature and the hop count hash value before replying by a RREP signed by the its private key. Similarly, the source and intermediate nodes have to verify the RREP to authenticate the identity of the sender. As RERR messages have a large amount of mutable information, SAODV suggests that all RERRs should be signed by the sender's private key.

S. Lee [89] proposed a solution that modified the AODV routing protocol by introducing two new packets; the route confirmation request (CREQ) and route confirmation reply (CREP). An intermediate node has to send a CREQ to its next-hop node toward the destination node in addition to a RREP to the source node. Upon receiving a CREQ, the next-hop node looks up its cache for a route to the destination. If it has a route, it sends the CREP to the source. After receiving the CREP, the source node can confirm the validity of the path by comparing the path in RREP and the one in CREP. If both agree, the source node judges that the route is appropriate. One drawback of this method is that it cannot avoid the cooperative blackhole attack if two consecutive nodes work together as the first node asked its next hop node to send CREP to the source.

M. Al-Shurman [31] developed a solution that requires additional computational overhead. A node maintains the sequence number for the last packet sent to the other nodes in one table and the sequence number for the last packet received from

the other nodes in a second table. Either an intermediate node or the destination node stores the sequence number of last packet received from the RREQ. Source node extracts the last packet sequence number when it receives RREP and compares it to the most recent value saved in its table. If matching occurs, the transmission takes place otherwise this forwarding node is considered as malicious node and an alarm message is sent to the other nodes to block this malicious node.

S. Kurosawa [90] introduced a mechanism that monitors the characteristic change of a node within a given time. A node monitors the characteristics of another node by observing the number of sent RREQs and the number of received RREPs and the mean destination sequence numbers of RREQs and RREPs. A node is recognized as blackhole node if its characteristics are changed over the monitoring period. Threshold value is compared to the difference between the monitored RREQ and RREP sequence numbers and nodes that have high difference between these two numbers are isolated from the network. Determining optimal threshold is the major drawback of this algorithm which can lead to the isolation of an innocent node as if it is a blackhole.

L. Tamilselvan [91] proposed a solution that is designed upon a Fidelity Table in which each participating node is assigned with a fidelity level that determines the node reliability. A default fidelity level is assigned to each node and this level is updated based on the behaviour of the node. When a source node receives RREP, it waits until receiving further route replies from its neighbouring nodes and then selects a neighbour node with a highest fidelity level to forward data to the destination node. A destination node acknowledges receiving the data by sending ACK. Updating the fidelity level of node relies on trusted participation of the node in the network. The source node increments or decrements the fidelity level of the forwarding node upon receiving or missing the ACK respectively. A node is eliminated from the network if its fidelity level reaches zero and the node marked as malicious. This technique is used to identify cooperative blackhole attacks for a safe route discovery. The main drawback of this solution is the high end-to-end delay, especially when the malicious node is far away from the source node.

P. Raj [92] developed an algorithm that adds an additional check for a node to accept RREP. The node tests if the RREP sequence number is higher than a threshold value. If the RREP sequence number is higher, the sending node is considered to be malicious and that node is added to the black list and this RREP is discarded. As soon as the node detects a malicious node, it sends an ALARM packet to inform its neighbours about this malicious and ignores all RREPs received from it. A node dynamically updates its threshold value each interval time as the average of the difference between the sequence numbers received in the RREP packets and their corresponding values in its routing table. The solution is designed mainly for single blackhole attack and does not detect cooperative blackhole attacks. Updating threshold value and forwarding ALARM packets increases the routing overhead. The wrong calculation of the threshold value may exclude an innocent node as if it is a blackhole.

N. Mistry [93] introduced a solution that depends on analysing all received RREPs. As source node receives the first RREP, it waits MOS_WAIT_TIME seconds to receive multiple RREPs. During this time, the source node saves all the received RREPs in a table. Thereafter, the source node analyses the stored RREPs from the table, and rejects any having a very high destination sequence number and marks the node as malicious. The remaining entries in the table are arranged according to their destination sequence number and the node with the highest number is selected. This technique also records the identity of suspected malicious nodes to discard any upcoming control packets received and/or forwarded from/to that node and a routing entry for that node will not be maintained. The algorithm introduces high end-to-end delay as nodes have to wait for multiple RREPs.

X. Li [94] proposed a Packet Forwarding Ratio (PFR) trust model. A trust value of a node depends on the PFR value which is the ratio of data packets forwarded to the data packets received. This trust value is incremented or decremented upon forwarding or dropping data packets respectively. Trust values ranges between 0 to 1; with 0 signifies malicious node and 1 signifies absolute trust. A node with low trust value is not allowed to forward data packets.

R. Vaghela [95] proposed a solution in which the source node forwards data packets to destination node through the first RREP. The source node stores other replies in Collect Route Reply Table (CRRT) which includes the sequence number and packet arrival time at CRRT table. Furthermore, the source checks CRRT for a common next hop node in the RREPs. If common next hop nodes found, it considers those paths to be safe.

T. Mahmoud [96] developed a detection technique called Intrusion Avoidance System (IASAODV). The proposed algorithm suggests that the source node has to wait a time before sending data, in order to receive multiple RREP messages. During this period, the source node stores the sequence number and the arrival time of all received RREP in a table. When the timer expires, then the proposed algorithm checks the number of RREP messages in the table. This algorithm assumes that only the destination node is the trusted node and receiving more than one RREP packet clarifies that one of these packets is created by the trusted destination node and the other messages are created by blackhole nodes.

N. Choudhary [97] introduced a solution that based on sensing the wireless channel. A node assigns a *max_trust* value to all its neighbouring nodes. The node excludes a neighbour whose trust value decreases less than *min_trust* from engaging in communication. When a node forwards a data packet, it sets a timer with it and listens to the wireless channel in promiscuous mode to ensure that this packet is forwarded by a next hop neighbour. When the timer expires without hearing the retransmission of this packet, the node reduces the trust value for its next hop node. Trust value information is updated and disseminated to other neighbouring nodes. If the trust value of a node decreases below *min_trust* value, it will be isolated by all the nodes in the network.

6.3 Anti-Blackhole (AB) Mechanism

AB is designed to mitigate the effect of the blackhole attack on the performance of MANET on-demand routing protocol. The mechanism uses the concept of Self-Protocol Trustiness (SPT) [88] in which detecting a malicious intruder is accom-

plished by complying with the normal protocol behaviour and luring the malicious node to give an implicit avowal of its malicious behaviour. The mechanism does not use cryptographic techniques which conserves the power and computation resources. The algorithm has no assumption about the malicious node behaviour which makes it robust and reliable. Furthermore, the mechanism neither adds new routing packets nor modifies the existing ones. Each node in the network has to monitor the behaviours of its neighbours to detect if any misbehaves as a blackhole node. Malicious nodes will be detected reliably within a short time. The algorithm guarantees 100% exclusion of only blackhole nodes and does not exclude innocent nodes that may forward a RREP. The only way for a malicious node to subvert the mechanism is to reply to a very small number of RREQ packets and therefore it cannot launch the blackhole attack or affect the network performance.

The idea is to record the rate of RREP packets received from a neighbour and use this rate to clarify if it will suspect on one or more of its neighbours or not. If the rate exceeds a threshold, the trustiness of this neighbour node is examined by sending a fake RREQ from a non-existent source node to a non-existent destination node. Only a malicious node will respond to this fake RREQ. If a node receives a RREP to its fake RREQ from a neighbour, the node becomes sure that this neighbour is a blackhole node, classifies it as malicious, and adds it to a black list of potential malicious nodes. Once on the black list, it will be removed from the routing table and any RREP from this black listed node are discarded, but they are still recorded. A node implementing the Anti-Blackhole mechanism behaves as follows:

- Every TRAFFIC_TIME, each node monitors the number of RREPs received from each neighbour since the last classification update was examined as shown in Algorithm 6.1 and Figure 6.1.
- RREP packets are processed normally when received from neighbours with a black_list value of 0.
- If the number of RREPs received from a neighbour exceeds the threshold RREP_THRESHOLD, this neighbour's black_list value is set to 1

Algorithm 6.1 AB Neighbour Classification

N: number of RREP received from a neighbour
T: RREP threshold
B: neighbour blacklist value (default assigned for normal node)

1: **for all** neighbour in list **do**
2: **if** $B \neq default$ **AND** $N > T$ **then**
3: increment B
4: **end if**
5: reset N
6: **end for**

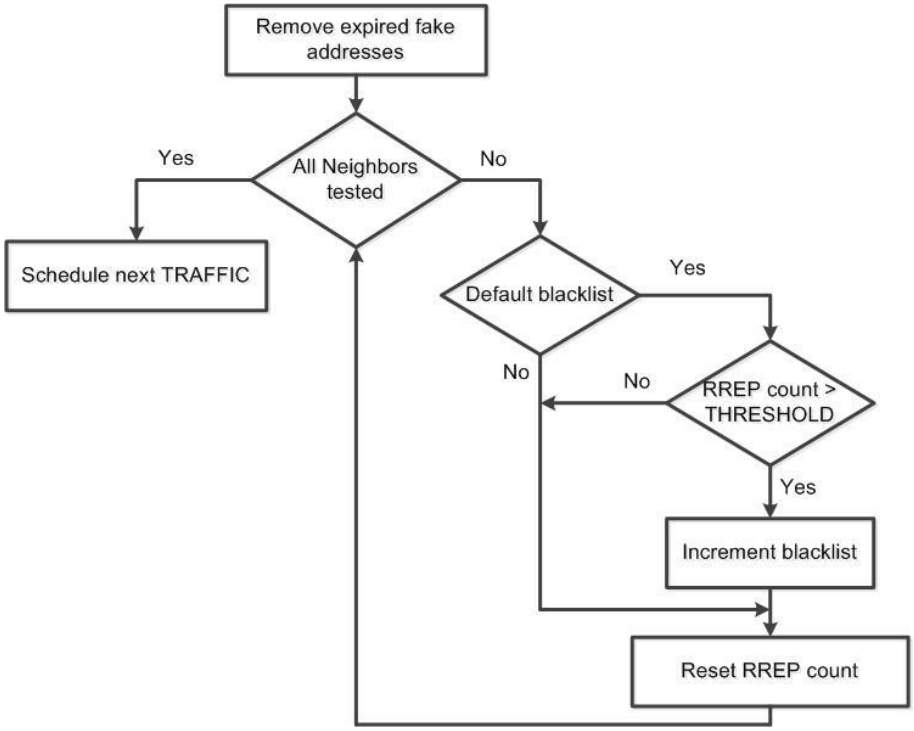

Figure 6.1: AB Neighbour Classification

which means that this neighbour is under suspicion. Choice of the RREP_THRESHOLD is made by running AODV hundreds number of scenarios in the absence of malicious nodes and observing the average number of RREPs that can be received during TRAFFIC_TIME. This choice of a low value of RREP_THRESHOLD leads to suspecting in many normal nodes but the node can differentiate later between a blackhole node and a normal node depending on their response.

- Each node counts the number of neighbours nodes that exceed this threshold during TRAFFIC_TIME.

- If the black_list value of neighbour is 1, the node tests the trustiness of its neighbour by sending a fake RREQ from a random non-existent source node to a random non-existent destination node. The node stores this fake source and destination to a trustiness table for later examination. The node also sets an expiry time for this entry to avoid table inflation. The node schedules a timer until RREP_VALIDATE seconds which is the testing duration to receive a RREP for this fake RREQ. To control the number of tests, the node sends fake RREQs as long as the number of these fake requests is less than the black listed nodes divided by RREP_LIMIT. This ratio decreases the routing overhead results from sending unrequired fake RREQs.

- To avoid flooding the network with fake RREQs which increases the routing overhead, the node sets the TTL value of this request to a small number. If TTL is set to 1, this implies that this fake RREQ will be received only by the neighbours of the node and all normal neighbours will drop this RREQ without forwarding it as the TTL value reaches zero and they have not any route to this fake destination while the blackhole node only replies to this fake RREQ.

- Algorithm 6.2 and Figure 6.2 show that if a RREP is received from a neighbour for this fake RREQ and both fake source and destination addresses are found in the trustiness table and the number of hops identifies that RREP originator is the neighbour (i.e. number of hops is 2), the node identifies that the originator is a blackhole node by incrementing its black_list value to 2, removes it from the routing table and drops any upcoming RREPs received from this neighbour without processing.

- If a RREP is received from a neighbour for the fake RREQ and both fake source and destination addresses are found in the trustiness table but the number of hops is greater than 2 which implies that this neighbour is forwarding a RREP originated from a blackhole node, the node drops this RREP.

Algorithm 6.2 AB RREP Processing

N: number of RREPs received from a neighbour
B: neighbour blacklist value (default assigned for normal node)

```
1: receive RREP
2: increment N
3: if B = exclusion then
4:     malicious misbehaviour
5:     discard RREP
6: else if B ≠ default AND reply for fake RREQ then
7:     reset N
8:     increment B
9:     discard RREP
10: else
11:     normal RREP processing
12: end if
```

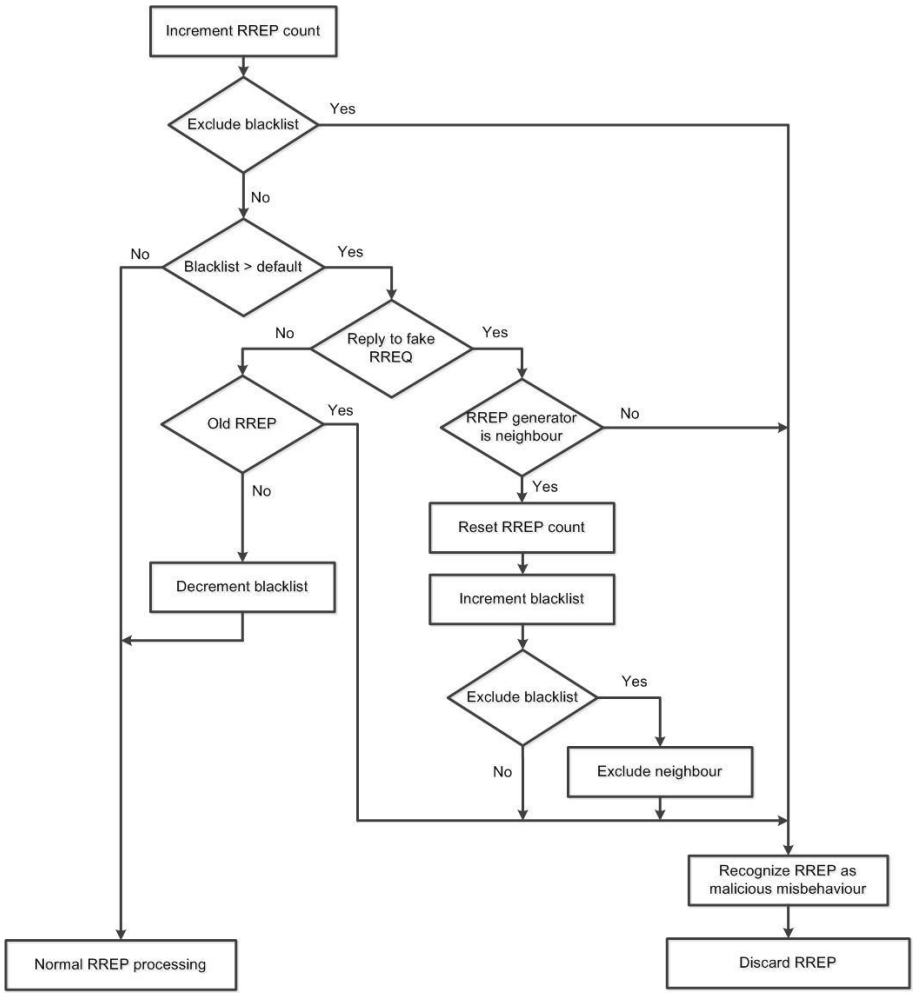

Figure 6.2: AB RREP Processing

- To avoid permanent blacklisting of an innocent node, the node resets the black_list value to 0 if a RREP_VALIDATE timer expires without receiving a reply to the fake RREQ.

Our proposed mechanism (AB) clarifies that each node is responsible for monitoring its neighbours and detecting and excluding blackhole ones. So, our mechanism does not differentiate between single and cooperative blackhole attack. This is because if the node at each terminal of a chain of cooperative blackhole nodes is detected by its neighbours, the chain becomes useless and cannot affect the network. Table 6.1 shows the values of parameters that were used in our simulations.

Table 6.1: AB Mechanism Parameters

RREP_THRESHOLD	2
RREP_LIMIT	3
RREP_VALIDATE	5 s
TRAFFIC_TIME	20 s

A malicious node can subvert the mechanism by sending only 2 RREPs at most to a neighbour every 20 seconds and therefore it cannot launch the blackhole attack or affect the network performance.

AODV controls flooding of RREQs by enforcing a source node that has not fresh route to a destination to use an expanding ring search technique as an optimization. In an expanding ring search, the source node initially sets the TTL value to TTL_START in the RREQ packet and if it does not receive a RREP it increments the TTL value by TTL_INCREMENT until the TTL value reaches TTL_THRESHOLD, beyond which it sets the TTL value to NET_DIAMETER is used for each flood. Although AODV suggests a value 1 for TTL_START [98], the NS-2 simulator version of AODV uses the value 5. So, setting the TTL value in the fake RREQ to 1 is compatible with the original AODV and has not a big difference with the simulator version. A malicious node that decided not to reply to all RREPs with TTL value of 1 to avoid detection will also not reply to genuine RREQs which limits the effectiveness of the blackhole attack. In addition, as a result of dropping RREP with hop count greater than 2 to the fake RREQ, the mechanism forces blackhole neighbours to use a hop count value 2 in all their fake RREPs which exposes it to detection if the RREQ is fake or sets their hop count greater than 2 to avoid detection and gives up claiming that they have best routes to destinations which exposes its reply to be dropped if the RREQ is fake.

Although AB mechanism succeeded in discovering malicious nodes within a small time with a ratio of 100% true exclusion of genuine malicious nodes, it has an obvious security gap. Disseminating the only threshold level used RREP_THRESHOLD introduces the ability for an attacker to subvert the algorithm by working below this threshold. Introducing a mechanism that overcome this drawback was the motivation to develop this algorithm by introducing the Blackhole Resisting Mechanism (BRM).

6.4 Blackhole Resisting Mechanism (BRM)

Blackhole Resisting Mechanism (BRM) [88] is designed to mitigate the effect of the blackhole attack on the performance of MANET reactive routing protocols by fast detection of blackhole neighbours. BRM modifies AB mechanism to close the security gaps and overcome its drawbacks such as using a fixed threshold RREP_THRESHOLD that can be disseminated to malicious nodes. The mechanism uses the concept of SPT which detects a malicious node by complying with the normal protocol behaviour and luring the malicious node to give an implicit avowal of its malicious behaviour. The mechanism does not use cryptographic techniques which conserves the power and computation resources. The algorithm makes no assumptions about blackhole behaviour which makes it robust and reliable. Furthermore, the mechanism neither adds new routing packets nor modifies the existing ones. We introduce a small modification to the original routing protocol by storing the last three per hop times for a RREP received for a destination. The per-hop time is calculated as the latency between sending a RREQ and receiving its corresponding RREP divided by the hop count value included in the RREP.

Each node in the network has to monitor the performance of its neighbours to detect if any misbehaves as a blackhole node. Malicious nodes will be detected reliably within a short time. The algorithm guarantees 100% exclusion of only blackhole nodes and does not exclude any innocent node that may forward a RREP. A malicious node cannot subvert the mechanism as there are no thresholds and all variables are randomly set.

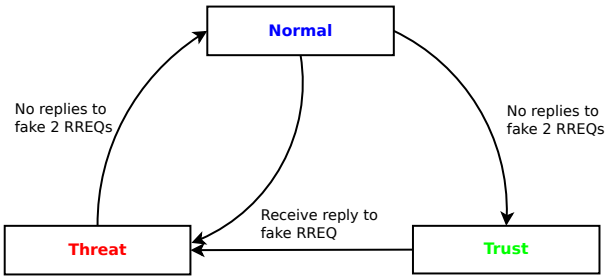

Figure 6.3: BRM Node Trust Level

The idea is to periodically send a fake RREQ from a non-existent source node to a non-existent destination node. Only a malicious node will respond to this fake RREQ. If a node receives a RREP to its fake RREQ from a neighbour, the node ensures that this neighbour is a blackhole node, declares it as malicious and adds it to a black list of potential malicious nodes. Once on the black list, it will be removed from the routing table. All RREPs from black listed nodes are not forwarded, but they are still recorded. The interval between two successive fake RREQs of a node is determined based on its neighbours' behaviours. The algorithm introduces two different variables; trust level and confidence level. The trust level is a node's trust in the network and its assessment of the instantaneous attack threat. Once a node participating in a network, it sets its trust level to ***Normal*** mode and it updates this trust level to either ***Trust*** or ***Threat*** upon reception of replies to its fake RREQs. Figure 6.3 shows the finite state machine diagram of the trust level modes of operations. The node assigns a confidence level to each neighbour. A confidence level for a neighbour is dynamically changed according to the neighbour behaviour to the testing node fake RREQ which consequently changes the testing node trust level. It is initialized to MAX_CONFIDENCE for a neighbour and decremented upon misbehaviour of this neighbour. A node implementing the Blackhole Resisting Mechanism behaves as follows:

- Algorithm 6.3 and Figure 6.4 show that a node periodically sends a fake RREQ from a random non-existent source node to a random non-existent destination node. The node stores these fake source and destination addresses in a trustiness table for later examination. The node also sets an expiry time for this entry to avoid the table inflation.

Algorithm 6.3 BRM Fake RREQ Scheduling

S: small interval time (***Threat***)
M: medium interval time (***Normal***)
L: large interval time (***Trust***)
T: next fake RREQ time
U: mode upgrade time (used if no RREP for fake RREQ)
C: confidence value of a neighbour
V: testing time to receive RREP for fake RREQ

```
1: send fake RREQ
2: if mode = Trust then
3:     T = randomly set from S
4: else if mode = Threat then
5:     T = randomly set from L
6: else
7:     T = randomly set from M
8: end if
9: schedule T
10: schedule U
11: C = T + V
```

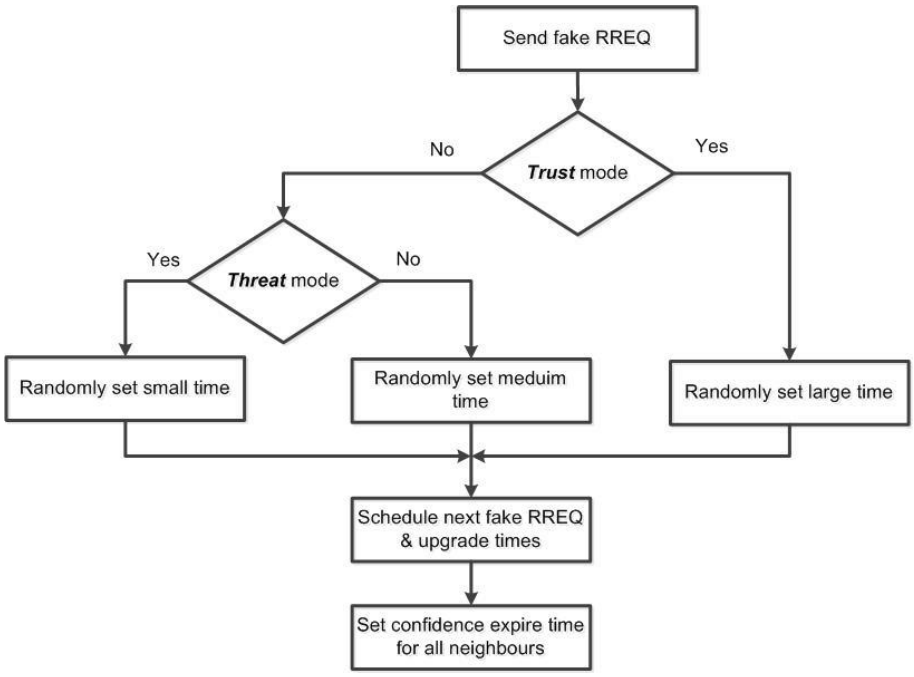

Figure 6.4: BRM Fake RREQ Scheduling

- A node usually sets its trust level to ***Normal*** and sends fake RREQs with a moderate rate at random time intervals between MIN_NORMAL and MAX_NORMAL. Once a node receives a reply for one of its fake RREQs, it changes its trust level to ***Threat*** and sends fake RREQs with a higher rate

at random time intervals between MIN_THREAT and MAX_THREAT. The node upgrades it trust level from ***Threat*** to ***Normal*** or from ***Normal*** to ***Trust*** if it sends two successive fake RREQs without receiving a reply during RREP_VALIDATE period. A node that set its trust level to ***Trust*** sends fake RREQs with a low rate at random time intervals between MIN_TRUST and MAX_TRUST. These three intervals introduce much more difficulty for a malicious node looking to subvert our proposed mechanism by tracing fake RREQs rate and differentiating it among genuine RREQs. To avoid overwhelming the network with unrequired routing overhead, our algorithm guarantees that at most one fake RREQ is sent by a node during MIN_TRUST seconds if there are no malicious nodes in the network which does not introduce a high routing overhead.

- To solve the trade-off between flooding the network with fake RREQs which increases the routing overhead and detection of validity of the RREQ by a malicious node, we suggest that the TTL value of this fake RREQ is set to a random number between MIN_TTL and MAX_TTL. We suggest values of 1 and 4 for these limits. Using a TTL of 1 ensures that any malicious neighbour will receive the fake RREQ. A limit of 4 ensures that this fake RREQ does not propagate too far in the MANET and hence overhead is limited. This limit also spoofs a malicious neighbour that the node which sent the RREQ to it is a forwarding node and it did not originate it for testing the malicious trustiness.

- Algorithm 6.4 shows that if a RREP is received from a neighbour for this fake RREQ and both fake source and destination addresses are found in the trustiness table and either the source address of this reply or the number of hops identifies that RREP originator is the neighbour (i.e. number of hops is 2), the node identifies that the originator is a blackhole node by setting its black_list value to 1, removes it from its routing table, and drops any upcoming RREPs received from this neighbour without processing. This check forces any malicious node attempting to launch a blackhole attack to claim

Algorithm 6.4 BRM RREP Processing

N: number of RREPs received from a neighbour
H: average latency of RREPs
L: latency of a RREP
B: neighbour blacklist value (default assigned for normal node)
C: confidence value of a neighbour
M: minimum confidence (neighbour is colluded node)
T: Time of next fake RREQ (depend on **Trust** level)

```
1: receive RREP
2: increment N
3: if B ≠ default then
4:     malicious misbehaviour
5:     discard RREP
6: else if reply for fake RREQ then
7:     if neighbour is RREP originator OR C = M then
8:         switch to Threat mode
9:         randomly set T
10:        increment B
11:    else if C > M then
12:        calculate L
13:        if L then < H
14:            decrement C
15:        end if
16:    end if
17:    discard RREP
18: else
19:    normal RREP processing
20: end if
```

that it only forwards this RREP by setting the hop count value greater than 2. This guarantees that any malicious neighbour will stop claiming it has best route to a destination by setting its reply hop count to 2.

- Figure 6.5 shows that if a RREP is received from a neighbour for this fake RREQ and both fake source and destination addresses are found in the trustiness table and the source address of this reply is not identical to the forwarding neighbour and the number of hops is greater than 2. This implies that this neighbour may be an innocent node that is used to forward this RREP or a malicious node that tries to subvert our algorithm. The node drops this RREP and computes the latency between sending the corresponding RREQ and this RREP and then divided this value by the hop count received in this RREP to calculate the per hop time for the received RREP. Then, the node compares

this value to the average hop time of all routes included in the routing table taking into account that each route has three previously stored per hop time values. If the per hop time of the received RREP is less than the average per hop time of all stored routes in the routing table, the node decrements this neighbour confidence level for each received RREP to a fake RREQ. The node clarifies that this neighbour attempting to use the blackhole characteristic of replying without checking its routing table which is a reason for receiving the RREP faster than those from other normal nodes.

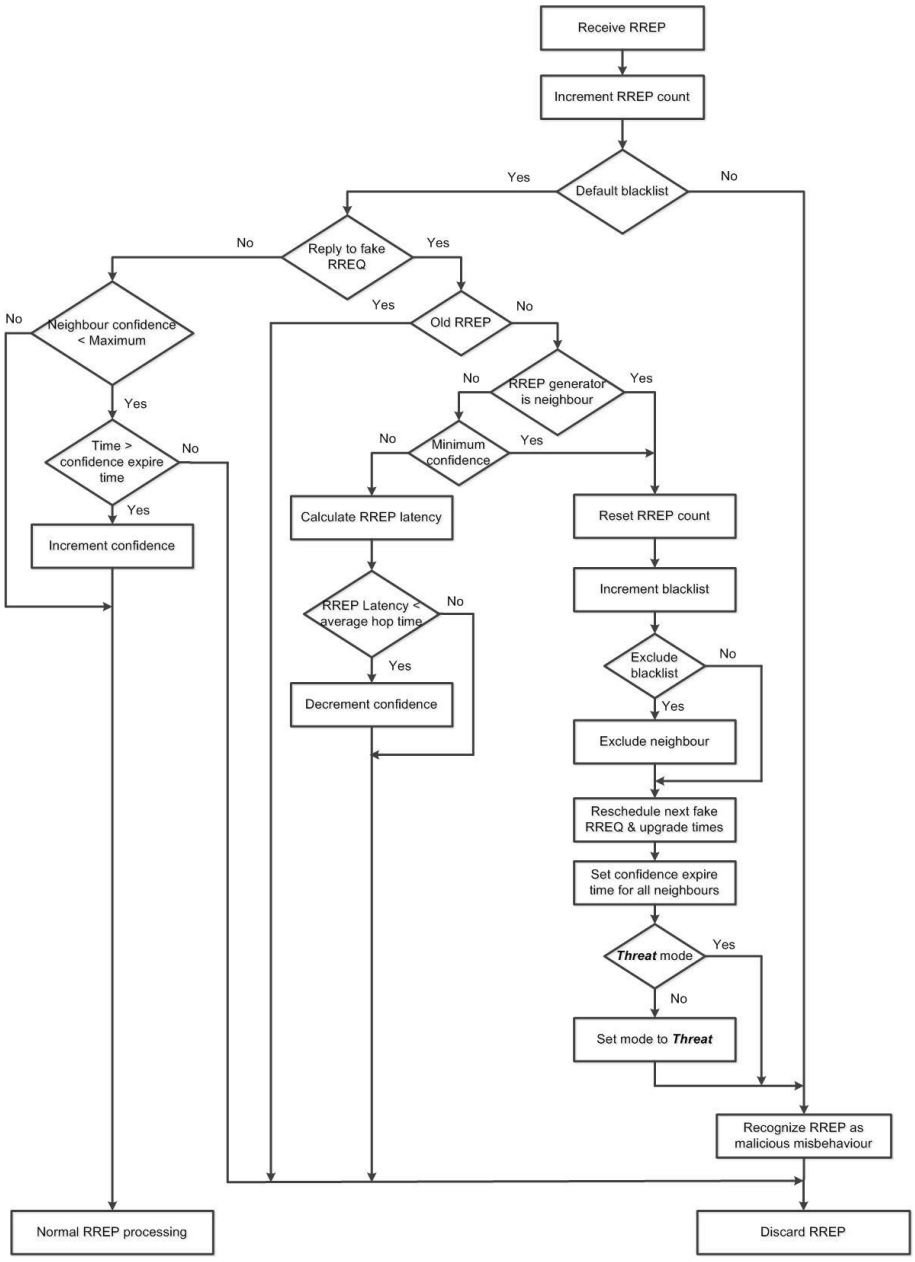

Figure 6.5: BRM RREP Processing

- If a neighbour confidence level becomes zero, a node identifies that this neighbour is a blackhole node or a colluding node. If this neighbour is not a malicious, it might be a colluding node as it should detect its malicious neighbour that uses it as victim node to forward RREPs. Decrementing a confidence level for a neighbour ensures that a node gives plenty of time for it to discover its malicious neighbours. So, a node sets the neighbour's black_list value to 1, removes it from its routing table and drops any upcoming RREPs received from this neighbour without processing.

Our proposed mechanism (BRM) clarifies that each node is responsible for monitoring its neighbours and detecting and excluding blackhole ones. So, our mechanism does not differentiate between single and cooperative blackhole attacks. This is because if the node at each end of a chain of cooperative blackhole nodes is detected by its neighbours, the chain becomes useless and cannot affect the network. Table 6.2 shows the values of parameters that were used in our simulations.

Table 6.2: BRM Mechanism Parameters

MAX_CONFIDENCE	3
MIN_TTL	1
MAX_TTL	4
RREP_VALIDATE	5 s
MIN_THREAT	5 s
MAX_THREAT	30 s
MIN_NORMAL	30 s
MAX_NORMAL	90 s
MIN_TRUST	90 s
MAX_TRUST	150 s

A malicious node that decided not to reply to all RREQs with TTL value between 1 and 4 to avoid detection will not reply to genuine RREQs and for sure all its fake replies to RREQ with higher TTL values are useless as a normal RREP will often be received by a source before its fake RREP. In addition, as a result of excluding nodes that replies by their own address or impersonating with hop counts value of 2, the mechanism forces malicious nodes to stop replying with their true identities and stop claiming that they have best routes. Moreover, calculating the per hop time of RREP with hop count greater than 2; the mechanism enforces blackhole neighbours

to delay their RREP which give RREQ sources an opportunity to receive replies from genuine nodes before the blackhole neighbour RREP. So, a malicious node follows our proposed mechanism either takes the risk to misbehave by sending fake RREP which increases the possibility of discovering it by its neighbour or stops misbehaving.

The BRM algorithm includes strong features that do not allow malicious nodes to subvert it. BRM has no thresholds, such as RREP rate, that may be disseminated to malicious nodes and introduce a way for these malicious nodes to work under these thresholds. Instead of this, the algorithm sends fake RREQs periodically at a rate depending on the trust level. The adjacent sending interval times of the three level of trustiness in addition to the normal RREQ of the original routing protocol introduces a difficulty for a malicious node to differentiate between genuine and fake RREQs. As shown from Table 6.2, a node sends fake RREQ randomly between 5 seconds and 150 seconds which is a long interval that makes the process of tracing fake RREQs complex. In addition, BRM introduces two limits MIN_TTL and MAX_TTL in which the TTL value has to be chosen randomly to introduce much more difficulty for a malicious node to estimate if the RREQ is genuine or one for testing. Moreover, BRM classifies its reaction to a reply for a fake RREQ based on the hop count value received in this RREP into two levels. The first if the hop count value is 2 which implies implicit recognition from a malicious neighbour. The second if the hop count value is greater than 2 which means that the forwarding neighbour may be a victim to a malicious node. The node decrements this neighbour confidence level until it reaches zero; when the node will be considered as a malicious neighbour. So, the worst scenario permitted by BRM algorithm is that a node accepts a limited number of RREPs (i.e. MAX_CONFIDENCE) to its fake RREQs from a neighbour; after which the node is certain that this neighbour is a malicious node.

As we will see later in simulations, BRM algorithm succeeded in detecting the majority of malicious neighbours in a short time. Detection of a malicious node is done whenever it replies to a fake RREQ, and because the generations of this fake RREQ is done randomly within a range (5 - 150 seconds), just few minutes are more

than enough to exclude malicious nodes from the network. BRM algorithm does not assume the attacker has to continue its malicious behaviour, but it guarantees that whenever a malicious node starts it bad behaviour, it will discovered immediately and this can be done within at most 150 seconds from beginning of its bad behaviour. So, a malicious node has two choices if it cannot differentiate between genuine and fake RREQs; either to reply to some or all RREQs and expose itself for detection and exclusion or to not reply to any RREQ to be safe from detection which leads to the same result of preventing blackhole attack. Our proposed algorithm does not care about all nodes in the MANET; each node only cares about its neighbours (1 hop only from it). If a node sends a fake RREQ and later receives a reply for it from one of its neighbour; it will be sure that this neighbour is a malicious or a colluded node and as a result the neighbour should be excluded. If each node succeeded in excluding its malicious neighbours; this for sure guarantees malicious-free routes.

6.5 Simulation Approach

NS-2 simulator [69] is used to simulate blackhole attack. The simulation is used to analyse the performance of the networks under the blackhole attacks with and without our new two mechanisms AB and BRM. The parameters used are shown in Table 6.3. While we examined our proposed mechanisms on both UDP and TCP traffic and the mechanisms succeeded in detecting blackhole neighbours and enhancing the network performance for both traffic, the chapter is focused on the results of the proposed mechanisms on the TCP traffic only. We examined our proposed mechanisms for different number of nodes (25, 50, 75 and 100) and different node speeds (0, 5, 10, 15, 20 and 25 m/s). The highest negative impact of malicious nodes usually appears on static networks and this effect decreases as node mobility increases [77], so we report here the case of static networks. Similarly, only the case of 100 node networks is reported, corresponding to a high density of nodes. This gives malicious nodes a high number of neighbours. We choose a large simulation time to ensure that most of the malicious nodes have been detected especially for scenarios with a large number of malicious nodes.

Table 6.3: Resisting Blackhole Attacks Simulation Parameters

Simulation Time	600 s
Simulation Area	1000 m^2
Number of Nodes	100
Number of Connections	150
Number of Malicious Nodes	0 - 10
Node Speed	0 - 30 m/s
Pause Time	10 s
Traffic Type	TCP

Our blackhole attack model assumes that once a malicious node receives a RREQ packet from a node, it immediately constructs a fake RREP that includes a randomly generated hop count between 2 and 4 to spoof other nodes about best route; i.e. 1 to 3 hop counts only from the RREQ source. The attacker assigns the destination sequence number value of this fake RREP as equal to the received one in the RREQ plus randomly generated number between 10 and 30 to spoof other nodes about the freshness of this RREP. Then, the attacker unicasts this fake RREP toward the RREQ source. A malicious node initiating a blackhole attack generates a fake RREP for each RREQ it receives to incorporate itself in all routes, therefore all packets are sent to a point where they are not forwarded anywhere which is a form of a DoS attack.

On the other hand, a testing node periodically sends a fake RREQ. Our model to generate this fake request assumes that the node randomly generates the source, the destination and their sequence numbers. It generates also TTL value for this request to control the distance that is traversed in the network and this value is randomly set between 1 and 4.

6.6 Simulation Results

AB and BRM algorithms achieve 100% success in excluding only blackhole neighbours without falsely excluding innocent nodes during simulations. AB algorithm is faster than BRM in detecting malicious neighbours and succeeded to discover more than 90% of malicious neighbours in few minutes. Although AB mechanism is faster than BRM in detecting blackhole neighbours, both mechanisms achieve very

similar results of the performances of the protocols. So, we compare their success ratio in excluding malicious nodes and then include only BRM effect on the network performance in this chapter. Details of the simulations are presented in the following sections. As mentioned earlier in Chapter 4, SAODV has a high resistance to blackhole attack. So, we do not incorporate our algorithms to it, instead we use its performance to compare with the corresponding non-secured protocol, i.e. AODV, in the presence and absence of our algorithm BRM.

As we mentioned earlier in Chapter 5, we use two parameters for measuring the exclusion efficiency of our algorithms. The first parameter is the total number of neighbour exclusions during the simulation time. The second parameter is the malicious discovery ratio which represents the ratio of malicious nodes discovered as time progresses to the total number of malicious nodes that should be discovered. In evaluating blackhole resisting mechanisms, we did not use the true exclusion ratio which represents the ratio of successful exclusion of genuine malicious nodes to the total number of exclusions as our algorithms achieves 100% success in excluding the malicious nodes only.

From the discussions presented earlier in Chapter 4, the blackhole attack has severe collapse on AODV and small negative impacts on DSR and AOMDV. Results show that BRM introduces huge improvements on AODV performance while it has slightly smaller degradation on both DSR and AOMDV. This clarifies that the degradation of the performance of our mechanism on DSR and AOMDV has no significant impact.

6.6.1 Resisting Blackhole Attacks in AODV

In this section, we evaluate the performance of AODV under blackhole attacks with and without our two mechanisms and compare with the results of SAODV. Our simulation shows that regardless of the number of nodes and the number of malicious nodes in the network, a node will detect a malicious neighbour within a short time. Figure 6.6 and Figure 6.7 show the proportion of malicious nodes that have been detected as time progresses for both AB and BRM algorithms

respectively. For clarity, we only show the results for 1, 4, 7 and 10 malicious nodes. The figures show that as the simulation time increases both the mechanisms succeeded in detecting and excluding malicious nodes. They also show that AB mechanism is faster than BRM and succeeded in excluding the vast majority of the blackhole neighbours in very short time. AB succeeded in excluding the majority blackhole neighbours after 120 seconds from the beginning of the simulation because most of the genuine RREQs and RREPs are sent during this period. Then the mechanism continues to exclude more malicious neighbours after that at a low rate. On the other hand, BRM has a continuous rate of detecting blackhole neighbours as it has not a threshold.

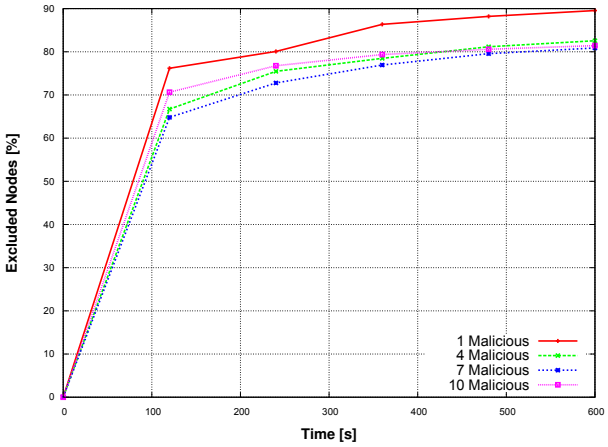

Figure 6.6: AB-AODV Malicious Discovery Ratio

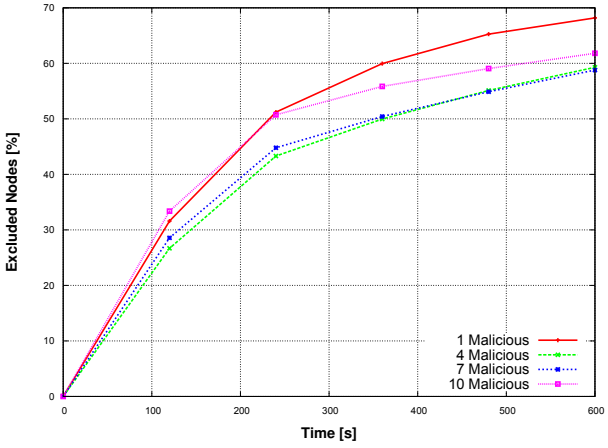

Figure 6.7: BRM-AODV Malicious Discovery Ratio

Figure 6.8 shows the total number of neighbours excluded during the simulation. The figure shows that AB excludes a larger number of malicious neighbours than BRM.

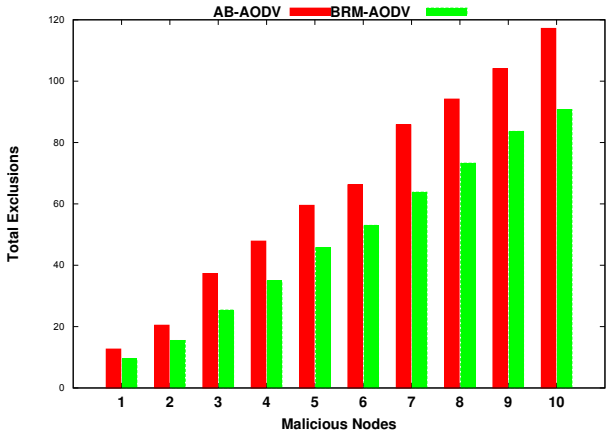

Figure 6.8: AODV Total Exclusions

The effect of BRM algorithm on the AODV packet delivery ratio is shown in Figure 6.9. While the blackhole attack has severe impact on the PDR of AODV especially for large number of malicious nodes, BRM-AODV introduces an approximately constant PDR regardless of the number of malicious nodes. On the other hand, while SAODV has a constant PDR regardless of the number of malicious nodes such as our algorithm; our algorithm achieves a better PDR value than SAODV.

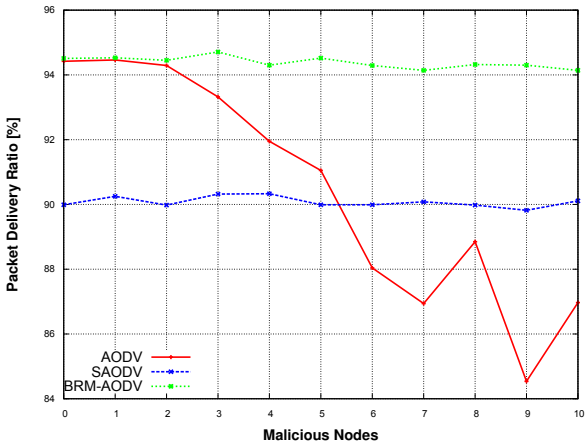

Figure 6.9: BRM Impact on AODV Packet Delivery Ratio

Figure 6.10 shows the effect of BRM algorithm on AODV network throughput. Throughput of BRM-AODV is better than AODV by approximately 25% for each

malicious node and the enhancement becomes huge for a high number of malicious nodes. While the throughput of AODV dramatically decreases as the number of malicious nodes increases, BRM-AODV slightly decreases for a high number of malicious nodes. In addition, our algorithm introduces a better throughput compared to SAODV of approximately 100%.

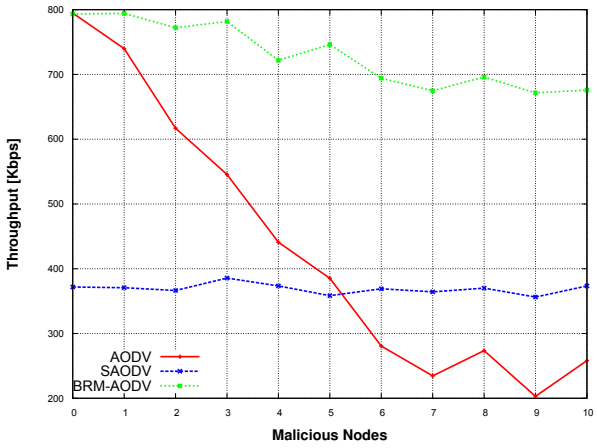

Figure 6.10: BRM Impact on AODV Network Throughput

The effect of BRM algorithm on the AODV end-end-delay is shown in Figure 6.11. The delay of the original AODV protocol is reduced as the number of malicious nodes increases which is slightly paradoxical as the attack improves the delay. As discussed earlier in Chapter 4, this is because the delay is only measured on packets that reach their destinations and since the blackhole nodes drop all received data routed through them, the number of packets that will be considered in calculating the delay decreases as the number of malicious nodes increases. As our proposed mechanism succeeded in receiving more data packets than AODV, the number of packets that will be considered in calculating the delay increases approaching the level that is the delay of network in the absence of malicious nodes. While the number of malicious nodes has very little effect on the delay in SAODV, our algorithm decreases it slightly further.

Figure 6.12 shows the effect of BRM algorithm on the AODV normalized routing load. The result shows that while the normalized routing load of AODV increases

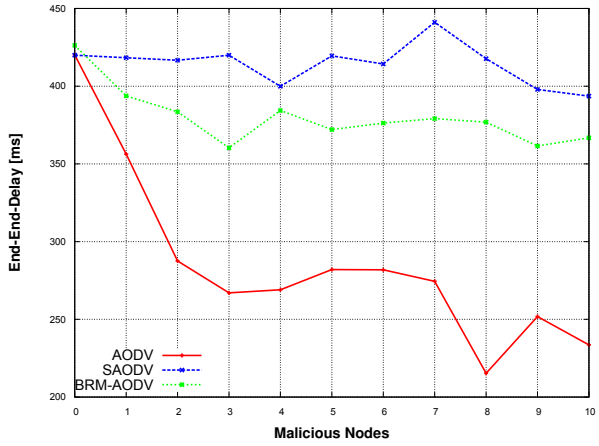

Figure 6.11: BRM Impact on AODV End-to-End Delay

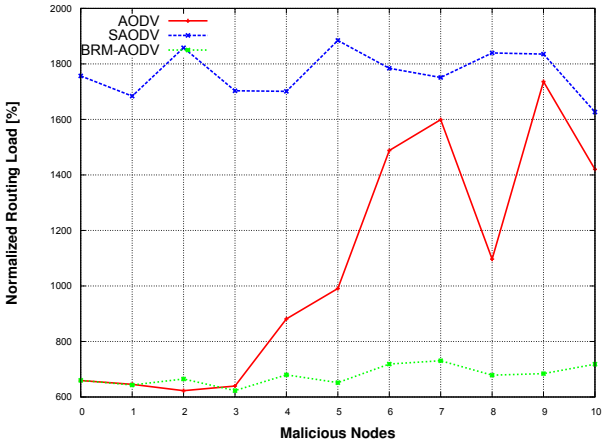

Figure 6.12: BRM Impact on AODV Normalized Routing Load

as the number of malicious nodes increases especially for large number of malicious nodes, it has not significantly changed for BRM-AODV. In addition, our algorithm introduces a better NRL than SAODV by approximately 300%.

Figure 6.13 shows the effect of BRM algorithm on the AODV routing overhead. The routing overhead of AODV decreases as a result of malicious nodes increases which is slightly confusing as the blackhole attack improves the routing overhead. This is because the blackhole nodes stop rebroadcasting the RREQ which decreases the number of RREQ packets, one of the factors used to measure the routing overhead as discussed before in Chapter 4. The routing overhead of BRM-AODV slowly decreases as the number of malicious nodes increases and our algorithm has small

differences compared to the routing overhead of the network in the absence of malicious nodes as a result of continuous detection of blackhole nodes. On the other hand, while routing overhead in SAODV is constant regardless of the number of malicious nodes, our algorithm introduces a better routing overhead than SAODV. Although the figure shows that the number of routing packets of our algorithm is less than its value of SAODV by approximately 5%, this enhancement increases up to approximately 400% if we consider the difference between AODV and SAODV packet sizes.

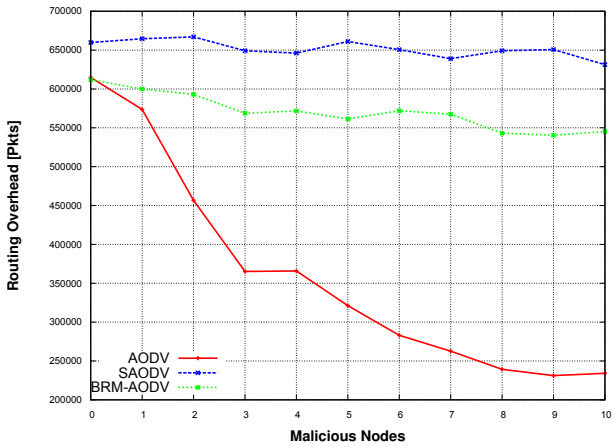

Figure 6.13: BRM Impact on AODV Routing Overhead

Figure 6.14 shows the effect of BRM algorithm on the AODV routing discovery latency. The result shows that RDL of AODV decreases dramatically as the number of malicious nodes increases which is slightly confusing as well that the blackhole attack improves RDL. This is because the fast response of blackhole nodes to RREQ which reduces the delay between sending a RREQ and receiving its corresponding RREP. The result shows that the routing discovery latency of BRM-AODV slowly decreases as the number of malicious nodes increases. In addition, our algorithm introduces a better RDL than SAODV by approximately 60%.

6.6.2 Resisting Blackhole Attacks in DSR

In this section, we compare the performance of DSR under blackhole attacks with and without our new mechanism. Figure 6.15 shows the proportion of malicious

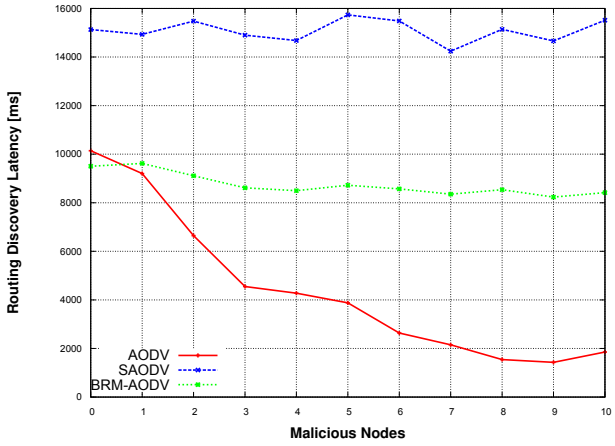

Figure 6.14: BRM Impact on AODV Route Discovery Latency

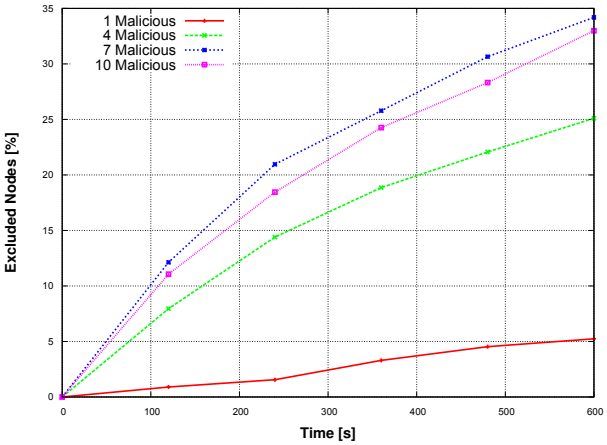

Figure 6.15: DSR Malicious Discovery Ratio

nodes that have been detected as time progresses. As mentioned earlier, we only show the results for 1, 4, 7 and 10 malicious nodes for clarity. The figure shows while our algorithm achieves a smaller ratio when incorporated to DSR than AODV, it is still succeeded in detecting and excluding malicious as the simulation time increases.

The effect of BRM algorithm on the DSR packet delivery ratio is shown in Figure 6.16. While BRM achieves slightly smaller PDR in absence of malicious nodes, it has a constant PDR regardless of the number of malicious nodes. On the other hand, the blackhole attack has a negative impact on the PDR of DSR especially for a large number of malicious nodes.

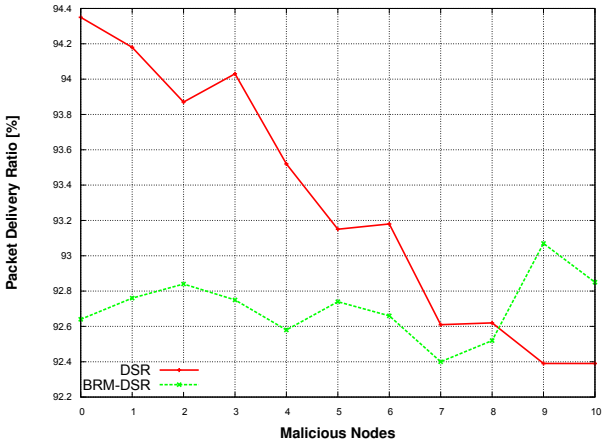

Figure 6.16: BRM Impact on DSR Packet Delivery Ratio

Figure 6.17 shows the effect of BRM algorithm on the DSR network throughput. Throughput of BRM-DSR achieves a nearly constant throughput regardless of the number of malicious nodes with a small degradation, approximately 15% on the average, compared to the original DSR. The figure shows also that the blackhole attack has a remarkable negative impact on the DSR throughput.

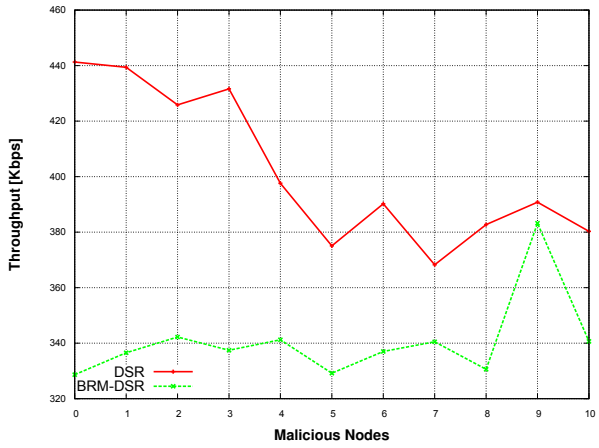

Figure 6.17: BRM Impact on DSR Network Throughput

The effect of BRM algorithm on the DSR end-end-delay is shown in Figure 6.18. The results show that the delay of DSR with and without our algorithm decreases as the number of malicious nodes increases which is slightly paradoxical as the attack improves the delay. This misleading result, which has been discussed earlier in Chapter 4, is a result of dropping received data routed through blackhole nodes. Incorporating our algorithm into DSR increases the delay of DSR by approximately

20% particularly when there are no malicious nodes. This is because that a node suspects to its neighbours and drops a high number of RREPs that exceeds its dynamic threshold until it ensures that its neighbours are innocents.

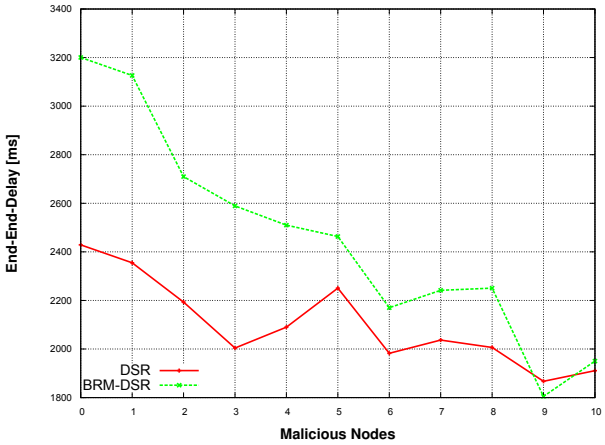

Figure 6.18: BRM Impact on DSR End-to-End Delay

Figure 6.19 shows the effect of BRM algorithm on the DSR routing overhead. The result shows that the routing overhead of DSR increases slightly as a result of malicious nodes increases. The result shows that the routing overhead of BRM-DSR has a small degradation as the number of malicious nodes increases. This is because as the number of malicious nodes increases, our algorithm increases the number of nodes under suspicious and hence increases the number of RREPs received from these suspicious nodes which decreases the number of RREQs forwarded by these suspicious nodes that leads reducing the routing overhead of the network.

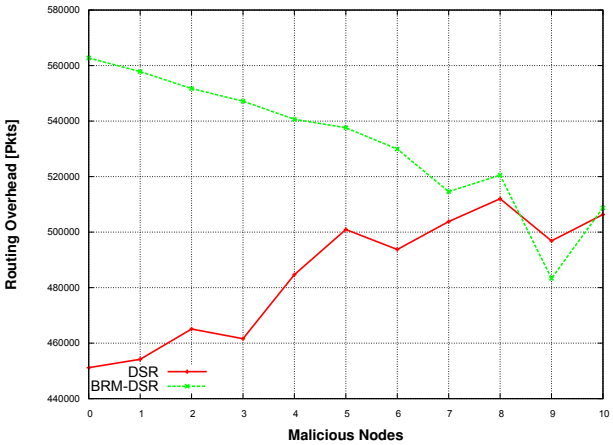

Figure 6.19: BRM Impact on DSR Routing Overhead

6.6.3 Resisting Blackhole Attacks in AOMDV

In this section, we compare the performance of AOMDV under blackhole attacks with and without our new mechanism. Figure 6.20 shows the proportion of malicious nodes that have been detected as time progresses. The figure shows while our algorithm achieves a smaller ratio when incorporated in AOMDV than AODV, it is still succeeded in detecting and excluding malicious as the simulation time increases.

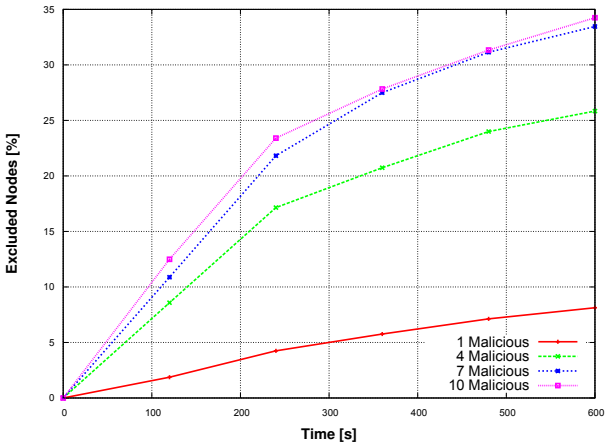

Figure 6.20: AOMDV Malicious Discovery Ratio

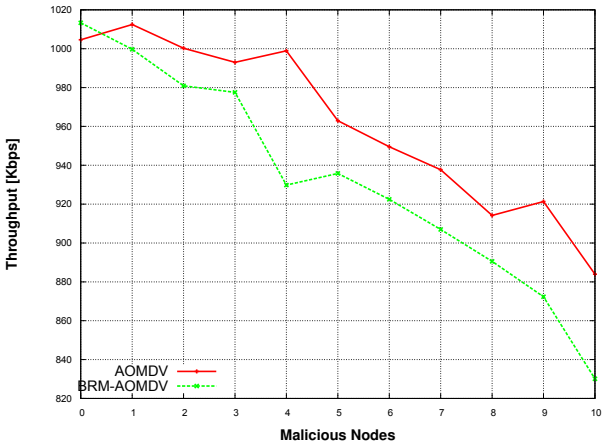

Figure 6.21: BRM Impact on AOMDV Network Throughput

Figure 6.21 shows the effect of BRM algorithm on the AOMDV network throughput. While the throughput of BRM-AOMDV has a negligible degradation when there is malicious nodes, compared to AOMDV, the throughput of both

protocols decreases as the number of malicious nodes increases. On the other hand, BRM achieves a small improvement in the throughput in the absence of malicious nodes.

The effect of BRM algorithm on the AOMDV end-end-delay is shown in Figure 6.22. While our algorithm increases the delay slightly compared to AOMDV especially for large number of malicious nodes, it achieves a small enhancement in the absence of malicious nodes. Delay of AOMDV with and without our mechanism decreases as the number of malicious nodes increases because of dropping data as discussed earlier in Chapter 4.

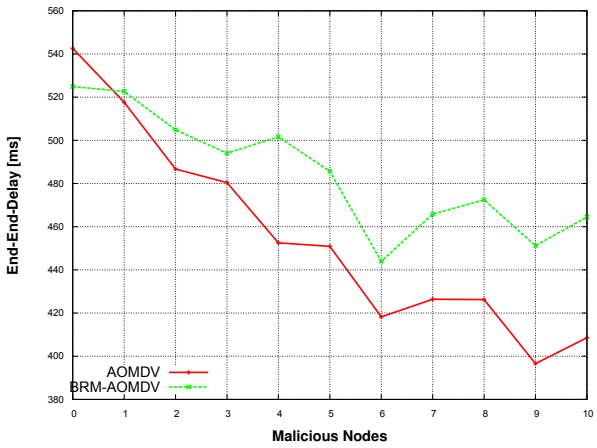

Figure 6.22: BRM Impact on AOMDV End-to-End Delay

Figure 6.23 shows the effect of BRM algorithm on the AOMDV routing overhead. The result shows that the routing overhead of AOMDV is approximately constant regardless of the number of malicious nodes. On the other hand, incorporating our algorithm into AOMDV has a dramatic effect on the overhead of AOMDV, cutting it by approximately 6% for each malicious node. This is because as the number of malicious nodes increases, the number of nodes under suspicious increases which increases the number of fake RREQs. In addition, AOMDV supports multiple RREQ copies which means that a RREQ is rebroadcast many times.

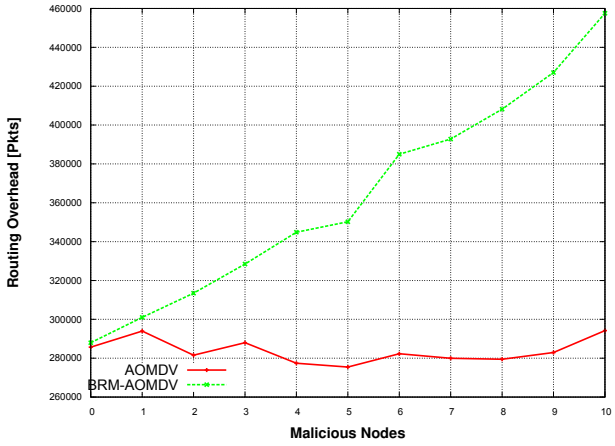

Figure 6.23: BRM Impact on AOMDV Routing Overhead

6.7 Summary

Blackhole attack has a dramatic impact on the MANET routing protocols performance. We proposed two new mechanisms Anti-Blackhole (AB) and Blackhole Resisting Mechanism (BRM) to resist blackhole attacks that can be incorporated into any reactive routing protocol. Both mechanisms use the concept of Self-Protocol Trustiness (SPT) in which detecting a malicious intruder is accomplished by complying with the normal protocol behaviour and luring the malicious node to give an implicit avowal of its malicious behaviour. BRM mechanism is designed to close the security gaps and overcome the drawbacks of AB mechanism that enable malicious nodes to subvert it. It succeeded in detecting and excluding a high ratio of blackhole nodes in a short time. It guarantees 100% exclusion of genuine blackhole nodes and does not exclude any innocent nodes that may forward a RREP. Neither solution requires expensive cryptography or authentication mechanisms or modifications to the packet formats.

Using NS2 simulation, we compare the performance of various protocols under blackhole attacks with and without our mechanisms, showing that it significantly reduces the effect of a blackhole attack. Although AB mechanism is faster than BRM at detecting blackhole neighbours, both mechanisms achieve very similar results in performances of the protocols. As the blackhole attack has severe collapse on AODV and small negative impacts on DSR and AOMDV. Results show that BRM

introduces huge improvements on AODV performance while it has slightly smaller degradation on both DSR and AOMDV which does not cause a significant negative impact.

Chapter 7

Conclusions

7.1 Introduction

MANETs are more prone to security threats than other wired and wireless networks because of their distributed system nature. Conventional MANET routing protocols assume that all nodes cooperate without maliciously disrupting the operation of the protocol and do not provide defence against malicious attackers. Current security solutions of the MANETs routing protocols can be mainly categorized into prevention mechanisms and detection and reaction mechanisms. We analysed and evaluated the performance of MANET reactive routing protocols in the presence of various attacks and introduced new mechanisms that resist some of them.

The rest of the chapter is organized as follows. Section 2 presents a summary of the thesis contributions. In Section 3, research limitations are highlighted. Section 3 suggests a future work.

7.2 Thesis Contributions

In this thesis, we have addressed the security problem of the MANETs reactive routing protocols. Selecting well-known reactive routing protocols that represent different approaches such as single path, multipath and secured from the wide span of these protocols require a detailed study of these protocols. Similarly, selecting multiple attacks that represent fabrications of routing packets, partial or full misbe-

having require a deep study of various attacks. The summary of the contributions that are presented in the thesis are as follows:

1. Analysing and evaluating the performance of MANET reactive routing protocols in the presence of various attacks. We demonstrated a result that the blackhole and flooding attacks have dramatic impact on the network performance. While the blackhole fabricates a fake reply, the flooding attack fabricates request which affects the network performance. On the other hand, grayhole and selfish attacks have a little negative impact on the performance of MANET routing protocols. We also discovered that the highest negative impact of malicious nodes in all these different attacks usually appears on static networks and this effect decreases as node mobility increases. Another important result is that while SAODV succeeded in resisting blackhole, grayhole and selfish attacks, it has performance degradation under the flooding attack. SAODV does not forward the routing packets without ensuring authenticity and integrity that explains its success in resisting blackhole, grayhole and selfish attacks. While SAODV achieves a moderate performance compared to the other protocols, its routing overhead is higher as a cost of its security features. SAODV cannot resist the flooding attack because a malicious node impersonating a non-existent node which could not be discovered by other non-malicious nodes. Finally, AOMDV achieves the lower negative impact to different attacks than other routing protocols.

2. Introducing a new concept in securing multi-hop networks such as MANETs called Self-Protocol Trustiness (SPT). This concept detects a malicious intruder by complying with the normal protocol behaviour and luring the malicious node to give an implicit avowal of its malicious behaviour.

3. Introducing an Anti-Flooding (AF) mechanism to resist flooding attacks which can be incorporated into any reactive routing protocol. It does not require expensive cryptography or authentication mechanisms, but relies on locally applied timers and thresholds to classify nodes as malicious. No modifications are made to the packet formats and hence it does not incur

any additional overhead. The proposed mechanism succeeded in discovering malicious nodes that attempt to flood the network regardless of the number of malicious nodes.

4. Introducing the Flooding Attack Resisting Mechanism (FARM) to resist flooding attacks. It has the main advantages of AF mechanism while it is designed to close the security gaps and overcome the drawbacks of AF mechanism. FARM mechanism uses our new concept of Self-Protocol Trustiness (SPT) to discover malicious intruders. It succeeded in detecting and excluding a high ratio of flooding nodes in a short time. FARM succeeded in detecting and excluding more than 80% of malicious neighbours in the simulation time with a highly trusted ratio exceeds than 90%.

5. Introducing the Anti-Blackhole (AB) mechanism to resist blackhole attacks which can be incorporated into any reactive routing protocol. It is designed based on our new concept of Self-Protocol Trustiness (SPT) in addition to some thresholds to discover a blackhole node. The algorithm neither adds new routing packets nor modifies the existing ones and it does not require expensive cryptography or authentication mechanisms. The algorithm guarantees 100% exclusion of only blackhole nodes and does not exclude any victim node that may forward a reply packet.

6. Introducing the Blackhole Resisting Mechanism (BRM) to resist blackhole attacks. It has the main advantages of AB mechanism while it is designed to close the security gaps and overcome the drawbacks of AB mechanism that enable malicious nodes to subvert it. It succeeded in detecting and excluding a high ratio of blackhole nodes in a short time.

From the previous discussion, it is clear that current security solutions for MANET reactive routing protocols can be classified as prevention mechanisms, and detection and reaction mechanisms. The disadvantages of the prevention mechanisms are their design dependency on the cryptographic algorithms which do not suit the MANET characteristics and their failure to discover some attacks. In addi-

tion, the detection and reaction mechanisms cannot guarantee the true classification of nodes because the cooperative nature of the MANETs which leads to false exclusion of innocent nodes and/or good classification of malicious nodes. Moreover, many solutions of both types are designed to suit a specific MANET routing protocol, as these solutions depend on the routing packet format of this protocol, which decreases the opportunity to use these solutions in other MANET routing protocols.

On the other hand, this thesis introduces a new concept of Self-Protocol Trustiness (SPT) to discover malicious nodes with a very high trustiness ratio of a node classification. This is because detecting a malicious intruder is based on an implied avowal of its malicious behaviour. SPT is used an underlying concept in designing different mechanisms to both flooding and blackhole attacks. From the above results, it is clear that using the SPT concept in both AB and RBM mechanisms ensures the fast discovery of malicious nodes within a very small time with 100% true exclusion of only malicious nodes. Although FARM mechanism achieves a very high true exclusion ratio exceeds than 90% compared to AF mechanism, and the previous solutions that did not consider this ratio, this ratio can be adapted later to achieve a complete success. The design and implementation of these mechanisms did not incorporate the use of cryptographic algorithms to suit the limited resources of the MANET nodes and did not depend on routing packet formats which increases their ability to be implemented in different MANET reactive protocols.

7.3 Future Work

This section highlights areas for potential future research, based on the contributions of this thesis. Many research ideas can be derived from our research work such as:

1. Extending the SPT concept to be applied in different wireless networks. Implementation of these new suggested algorithms, the BRM and the FARM algorithms, can be tested in other hop-by-hop networks and other wireless networks that do not have centralized structure such as sensor networks confirming their general applicability.

2. Introducing similar algorithms that can resist other attacks on MANETs using the SPT concept. SPT can be used as an underlying concept to design algorithms that can resist the other dangerous attacks such as wormhole attack.

3. Combining BRM and FARM algorithms to achieve a robust algorithm that can resist blackhole and flooding attacks simultaneously. Incorporating these algorithms in other various MANET proactive and reactive routing protocols to ensure their applicability.

4. Examining these new algorithms in larger networks to confirm the success of their design and to discover and address their advantages and disadvantages in these networks. This is can be a key to enhance the FARM algorithm to achieve success in discovering and excluding only genuine malicious nodes.

Appendix A

Modified Traffic Generator

```
#
#  This Tcl script generates a malicious node type scenario.
#  It accepts five arguments:
#     Type of connections (type)
#     Number of nodes (nn)
#     Number of connections (mc)
#     CBR packet rate/sec (rate)
#     Seed number (seed) [optional]
#
#  The command format is:
#
#       ns cbrgen.tcl -type cbr -seed 1 -nn 10 -mc 4 -rate 4.0
#
#  The code generates nn number of ns nodes and mc number of connections.
#  So, each node generation code in the format:
#
#     set udp_(0) [new Agent/UDP]
#     $ns_ attach-agent $node_(0) $udp_(0)
#     set null_(0) [new Agent/MyNull]
#     $ns_ attach-agent $node_(4) $null_(0)
#     set cbr_(0) [new Application/Traffic/CBR]
#     $cbr_(0) set packetSize_ 512
#     $cbr_(0) set interval_ 0.25
#     $cbr_(0) set random_ 1
#     $cbr_(0) set maxpkts_ 10000
#     $null_(0) drop-target [$node_(4) mobility-trace Drop "AGT"]
#     $cbr_(0) attach-agent $udp_(0)
#     $ns_ connect $udp_(0) $null_(0)
#     $ns_ at 1.00000 "$cbr_(0) start"
#
#  The code is created by Mohamed Abdelshafy
#
#  Copyright (c) 2014 by the Heriot-Watt University
#  All rights reserved.
#
# Connection Scenario construction code.
#
# $Header: .../ns-2/indep-utils/cbrgen.tcl,v 2.0 Mon 5/8/2014 12:40:00
```

```
# ======================================================================
# Default Script Options
# ======================================================================
set opt(nn)   0     ;# Number of Nodes
set opt(seed)   0.0
set opt(mc)   0
set opt(pktsize) 512

set opt(rate)   0
set opt(interval) 0.0   ;# inverse of rate
set opt(type)         ""

# ======================================================================

proc usage {} {
    global argv0

    puts "\nusage: $argv0 \[-type cbr|tcp\] \[-nn nodes\] \[-seed seed\]
        \[-mc connections\] \[-rate rate\]\n"
}

proc getopt {argc argv} {
   global opt
   lappend optlist nn seed mc rate type

   for {set i 0} {$i < $argc} {incr i} {
      set arg [lindex $argv $i]
      if {[string range $arg 0 0] != "-"} continue

      set name [string range $arg 1 end]
      set opt($name) [lindex $argv [expr $i+1]]
   }
}

proc create-cbr-connection { src dst } {
   global rng cbr_cnt opt

   set stime [$rng uniform 0.0 30.0]

   puts "#\n# $src connecting to $dst at time $stime\n#"
   puts "set udp_($cbr_cnt) \[new Agent/UDP\]"
   puts "\$ns_ attach-agent \$node_($src) \$udp_($cbr_cnt)"
   puts "set null_($cbr_cnt) \[new Agent/MyNull\]"
   puts "\$ns_ attach-agent \$node_($dst) \$null_($cbr_cnt)"
   puts "set cbr_($cbr_cnt) \[new Application/Traffic/CBR\]"
   puts "\$cbr_($cbr_cnt) set packetSize_ $opt(pktsize)"
   puts "\$cbr_($cbr_cnt) set interval_ $opt(interval)"
   puts "\$cbr_($cbr_cnt) set random_ 1"
   puts "\$cbr_($cbr_cnt) set maxpkts_ 10000"
   puts "\$null_($cbr_cnt) drop-target \[\$node_($dst) mobility-trace Drop
      \"AGT\"\]"
   puts "\$cbr_($cbr_cnt) attach-agent \$udp_($cbr_cnt)"
   puts "\$ns_ connect \$udp_($cbr_cnt) \$null_($cbr_cnt)"
```

```
   puts "\$ns_ at $stime \"\$cbr_($cbr_cnt) start\""

   incr cbr_cnt
}

proc create-tcp-connection { src dst } {
   global rng cbr_cnt opt

   set stime [$rng uniform 0.0 30.0]

   puts "#\n# $src connecting to $dst at time $stime\n#"
   puts "set tcp_($cbr_cnt) \[new Agent/TCP\]"
   puts "\$ns_ attach-agent \$node_($src) \$tcp_($cbr_cnt)"
   puts "set sink_($cbr_cnt) \[new Agent/MyTCPSink\]"
   puts "\$ns_ attach-agent \$node_($dst) \$sink_($cbr_cnt)"
   puts "set ftp_($cbr_cnt) \[new Application/FTP\]"
   puts "\$tcp_($cbr_cnt) set window_ 32"
   puts "\$tcp_($cbr_cnt) set packetSize_ $opt(pktsize)"
   puts "\$sink_($cbr_cnt) drop-target \[\$node_($dst) mobility-trace Drop
      \"AGT\"\]"
   puts "\$ftp_($cbr_cnt) attach-agent \$tcp_($cbr_cnt)"
   puts "\$ns_ connect \$tcp_($cbr_cnt) \$sink_($cbr_cnt)"
   puts "\$ns_ at $stime \"\$ftp_($cbr_cnt) start\""

   incr cbr_cnt
}

# ======================================================================

getopt $argc $argv

if { $opt(type) == "" } {
    usage
    exit
} elseif { $opt(type) == "cbr" } {
    if { $opt(nn) == 0 || $opt(seed) == 0.0 || $opt(mc) == 0 || $opt(rate)
       == 0 } {
   usage
   exit
    }

    set opt(interval) [expr 1 / $opt(rate)]
    if { $opt(interval) <= 0.0 } {
   puts "\ninvalid sending rate $opt(rate)\n"
   exit
    }
}

puts "#\n# nodes: $opt(nn), max conn: $opt(mc), send rate: $opt(interval),
    seed: $opt(seed)\n#"
set max_node [expr $opt(nn) - 0.5]

set rng [new RNG]
```

```
$rng seed $opt(seed)

set u [new RandomVariable/Uniform]
$u set min_ 0
$u set max_ $max_node
$u use-rng $rng

set cbr_cnt 0
set src_cnt 0

set srclist {}

while { $cbr_cnt < $opt(mc) } {

  set src [expr round([$u value])]
  set dst [expr round([$u value])]

  set y [lsearch $srclist $src]

  if { $dst == $src } {
    set dst [expr $dst + 1]
  }

  if { $cbr_cnt < $opt(nn) } {

    if { $y == -1 } {

      incr src_cnt

      lappend srclist $src

      if { $opt(type) == "cbr" } {
   create-cbr-connection $src $dst
      } else {
   create-tcp-connection $src $dst
      }

    }

  } else {

      if { $y == -1 } {

   incr src_cnt

   lappend srclist $src

      }

      if { $opt(type) == "cbr" } {
   create-cbr-connection $src $dst
      } else {
   create-tcp-connection $src $dst
```

```
      }

  }

}

puts "#\n#Total sources/connections: $src_cnt/$cbr_cnt\n#"
```

Appendix B

Node-Type Generator

```
#
#  This Tcl script generates a malicious node type scenario.
#  It accepts three arguments:
#     Total nuber of nodes (n)
#     Number of malicious nodes (m)
#     Attack type (a)
#     Attack time (t)
#     Seed number (seed) [optional]
#
#  The command format is:
#
#       ns nodes.tcl -n 10 -m 4 -a 3 -t 1 -seed 1
#
#  The code generates n number of ns nodes generation code and
#  randomly select if the node is malicious node or not.
#  So, each node generation code in the format:
#
#     set node_(0) [$ns_ node]
#     $node_(0) random-motion 0
#     $ns_ at 1.234567 "$node_(0) set malicious_ 1"
#     $ns_ at 1.234567 "$node_(0) set attack_ 3"
#
#
#  The code is created by Mohamed Abdelshafy
#
#  Copyright (c) 2014 by the Heriot-Watt University
#  All rights reserved.
#
#
# Node type construction code.
#
# $Header: .../ns-2/indep-utils/node.tcl,v 2.0 Mon 5/8/2014 12:40:00

# ======================================================================
# Default Script Options
# ======================================================================
set opt(n)    0     ;# Number of nodes
set opt(seed)   0.0
set opt(m)    0     ;# Number of malicious nodes
```

```
set opt(a)    0     ;# Attack type
set opt(t)    0     ;# Attack Random time

# ======================================================================

proc usage {} {
    global argv0
    puts stderr "\nusage: $argv0 \[-n nodes\] \[-m malicious\] \[-a
        attack\] \[-t time\] \[-seed seed\]\n"
}


proc getopt {argc argv} {
   global opt
   lappend optlist n m a t seed

   for {set i 0} {$i < $argc} {incr i} {
      set arg [lindex $argv $i]
      if {[string range $arg 0 0] != "-"} continue

      set name [string range $arg 1 end]
      set opt($name) [lindex $argv [expr $i+1]]
   }
}

proc create-normal-node { } {
   global node_cnt

   puts "#\n# creating node $node_cnt\n#"
   puts "set node_($node_cnt) \[\$ns_ node\]"
   puts "\$node_($node_cnt) random-motion 0"
   puts "\$node_($node_cnt) set malicious_ 0"
}

proc create-malicious-node { } {
   global node_cnt t x opt

   puts "#\n# creating node $node_cnt\n#"
   puts "set node_($node_cnt) \[\$ns_ node\]"
   puts "\$node_($node_cnt) random-motion 0"
   puts [format "\$ns_ at %.6f \"\$node_($node_cnt) set malicious_ %d\""
       $t $x]
   puts [format "\$ns_ at %.6f \"\$node_($node_cnt) set attack_ %d\"" $t
       $opt(a)]
}

# ======================================================================
# Main Program
# ======================================================================

getopt $argc $argv

if { $opt(n) == 0 } {
```

```
    usage
    exit
}

if { $opt(m) > $opt(n) } {
    puts stderr "error: The number of malicious node should less than the
        total number of nodes"
    exit
}

puts "#\n# nodes: $opt(n), malicious: $opt(m), attack: $opt(a), random
    time: $opt(t), seed: $opt(seed)\n#"

set nseed [expr $opt(seed) + 5]
set sseed [expr $opt(seed) * 3]

if {$opt(n) > 25} {
  if {$opt(m) > 0} {
    set minint [expr round($opt(n)/50)]
    set maxint [expr round($opt(n)/$opt(m))]
  } else {
    set minint $opt(n)/10
    set maxint $opt(n)
  }
} else {
  set minint 0
  set maxint 0
}

set maliciousRNG [new RNG]
$maliciousRNG seed $opt(seed)
set malicious [new RandomVariable/Uniform]
$malicious set min_ 0
$malicious set max_ 1
$malicious use-rng $maliciousRNG

set intervalRNG [new RNG]
$intervalRNG seed $nseed
set interval [new RandomVariable/Uniform]
$interval set min_ $minint
$interval set max_ $maxint
$interval use-rng $intervalRNG

set startRNG [new RNG]
$startRNG seed $sseed
set start_time [new RandomVariable/Uniform]
$start_time set min_ 0
$start_time set max_ 50
$start_time use-rng $startRNG

set malicious_cnt 0   ;# number of random malicious
set rndcrt 0       ;# randmly created nodes
```

```
for {set node_cnt 0} {$node_cnt < $opt(n)} {incr node_cnt} {

  if {$opt(m) == 0} {

    create-normal-node

  } else {

   set x [expr round([$malicious value])]

   if {$x == 1} {

       incr malicious_cnt

       if {$malicious_cnt <= $opt(m)} {

      if {$opt(t) == 1} {
        set t [$start_time value]
      }

      incr rndcrt
      create-malicious-node

      set y [expr round([$interval value])]

      for {set i 0} {$i < $y} {incr i} {

        incr node_cnt

        if {$node_cnt == $opt(n)} {
          set node_cnt [expr $node_cnt - 1]
          break
        }

        create-normal-node

      }

       } else {

         if {$node_cnt < $opt(n)} {
      create-normal-node
         }

       }
  }

   if {$x == 0} {

     if {$node_cnt < $opt(n)} {
       create-normal-node
     }
```

```
   }
  }

  set t 0
}

puts "#\n#Total Malicious/Nodes: $rndcrt/$node_cnt\n#"
```

Bibliography

[1] Ad hoc networks. `http://www.acorn.net.au/telecoms/adhocnetworks/adhocnetworks.html` [retrieved: April, 2016].

[2] Nelly Delgado. *Web Service Security: Scenarios, Patterns, and Implementation Guidance for Web Services Enhancements (WSE) 3.0*. Microsoft, USA, 2005.

[3] Priyanka Goyal, Sahil Batra, and Ajit Singh. A literature review of security attack in mobile ad-hoc networks. *International Journal of Computer Applications*, 9(12):11–15, November 2010.

[4] Monu Singh, Ajay Singh, Rajesh Tanwar, and Ritu Chauhan. Security attacks in mobile adhoc networks. *IJCA Proceedings on National Workshop-Cum-Conference on Recent Trends in Mathematics and Computing 2011*, RTMC(11), May 2012.

[5] Azzedine Boukerche, Begumhan Turgut, Nevin Aydin, Mohammad Z. Ahmad, Ladislau Bölöni, and Damla Turgut. Routing protocols in ad hoc networks: A survey. *Computer Networks*, 55(13):3032–3080, 2011.

[6] Ashwani Kumar. Security attacks in MANET - a review. *IJCA Proceedings on National Workshop-Cum-Conference on Recent Trends in Mathematics and Computing 2011*, RTMC(11), May 2012.

[7] Kamaljit I. Lakhtaria and Kamaljit I. Lakhtaria. *Technological Advancements and Applications in Mobile Ad-Hoc Networks: Research Trends*. IGI Global, Hershey, PA, USA, 1st edition, 2012.

[8] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. *SIGCOMM Comput. Commun. Rev.*, 24(4):234–244, October 1994.

[9] Shree Murthy and J. J. Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. *Mob. Netw. Appl.*, 1(2):183–197, October 1996.

[10] Guangyu Pei, Mario Gerla, and Tsu-Wei Chen. Fisheye state routing in mobile ad hoc networks. In *ICDCS Workshop on Wireless Networks and Mobile Computing*, pages 71–78, 2000.

[11] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized link state routing protocol for ad hoc networks. In *Proceedings of IEEE International Multi Topic Conference (INMIC) Technology for the 21st Century*, pages 62–68, 2001.

[12] David B. Johnson and David A. Maltz. Dynamic source routing in ad hoc wireless networks. *Mobile Computing*, 353:153–181.

[13] Charles E. Perkins and Elizabeth M. Royer. Ad-hoc on-demand distance vector routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, 1997.

[14] Chai-Keong Toh. Associativity-based routing for ad hoc mobile networks. *Wirel. Pers. Commun.*, 4(2):103–139, March 1997.

[15] Zygmunt J. Haas and Marc R. Pearlman. The performance of query control schemes for the zone routing protocol. *IEEE/ACM Trans. Netw.*, 9(4):427–438, 2001.

[16] Prasun Sinha Raghupathy, Prasun Sinha, Raghupathy Sivakumar, and Vaduvur Bharghavan. CEDAR: a core-extraction distributed ad hoc routing algorithm. *IEEE Journal on Selected Areas in Communications*, 17:1454–1465, 1999.

[17] Shuo Wang, Qiao Li, Yule Jiang, and Huagang Xiong. Stable on-demand multipath routing for mobile ad hoc networks. In *Computational Intelligence and*

Industrial Applications, 2009. PACIIA 2009. Asia-Pacific Conference on, volume 2, pages 318–321, Nov 2009.

[18] Mohammed Tarique, Kemal E. Tepe, Sasan Adibi, and Shervin Erfani. Survey of multipath routing protocols for mobile ad hoc networks. *Journal of Network and Computer Applications*, 32(6):1125–1143, November 2009.

[19] Mahesh K. Marina and Samir R. Das. Ad hoc on-demand multipath distance vector routing. *Wireless Communications and Mobile Ccomputing (WCMC): Special Issue: Wireless Ad Hoc Networks: Technologies and Challenges*, 6(7):969–988, November 2006.

[20] L. Reddeppa Reddy and S. V. Raghavan. Smort: Scalable multipath on-demand routing for mobile ad hoc networks. *Ad Hoc Netw.*, 5(2):162–188, March 2007.

[21] Rajib Das, Bipul Syam Purkayastha, and Prodipto Das. Security measures for black hole attack in MANET: An approach. *CoRR*, abs/1206.3764, 2012.

[22] K.P. Manikandan, R.Satyaprasad, and K.Rajasekhararao. A Survey on Attacks and Defense Metrics of Routing Mechanism in Mobile Ad hoc Networks. *IJACSA - International Journal of Advanced Computer Science and Applications*, 2(3):7–12, 2011.

[23] S. A. Razak, S. M. Furnell, N. L. Clarke, and P. J. Brooke. Friend-assisted intrusion detection and response mechanisms for mobile ad hoc networks. *Ad Hoc Netw.*, 6(7):1151–1167, September 2008.

[24] Issa Khalil. *Mitigation of Control and Data Traffic Attacks in Wireless*. PhD thesis, Purdue University, 05 2007.

[25] Marek Klonowski, Miroslaw Kutylowski, and Anna Lauks. Repelling detour attack against onions with re-encryption. In *Proceedings of the 6th International Conference on Applied Cryptography and Network Security*, ACNS'08, pages 296–308, Berlin, Heidelberg, 2008. Springer-Verlag.

[26] Pietro Michiardi and Refik Molva. Prevention of denial of service attacks and selfishness in mobile ad hoc networks. In *Mobile Ad Hoc Networks, Institut Eurecom Research Report RR-02-063*, 2002.

[27] Henk C. A. van Tilborg and Sushil Jajodia, editors. *Encyclopedia of Cryptography and Security, 2nd Ed.* Springer, 2011.

[28] Chris Piro, Clay Shields, and Brian Neil Levine. Detecting the sybil attack in mobile ad hoc networks. In *Proc. SecureComm*, pages 1–11. IEEE Press, 2006.

[29] Chi Zhang, Xiaoyan Zhu, Yang Song, and Yuguang Fang. A formal study of trust-based routing in wireless ad hoc networks. In *Proceedings of the 29th conference on Information communications*, INFOCOM'10, pages 2838–2846, Piscataway, NJ, USA, 2010. IEEE Press.

[30] Yinghua Guo and Sylvie Perreau. Detect DDoS flooding attacks in mobile ad hoc networks. *Int. J. Secur. Netw.*, 5(4):259–269, December 2010.

[31] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Rushing attacks and defense in wireless ad hoc network routing protocols. In *Proceedings of the 2nd ACM workshop on Wireless security*, WiSe '03, pages 30–40, New York, NY, USA, 2003. ACM.

[32] Mohammad Al-Shurman, Seong-Moo Yoo, and Seungjin Park. Black hole attack in mobile ad hoc networks. In *Proceedings of the 42nd annual Southeast regional conference ACM-SE 42*, pages 96–97. ACM Press, April 2004.

[33] Megha Arya and Yogendra Kumar Jain. Grayhole attack and prevention in mobile adhoc network. *International Journal of Computer Applications*, 27(10):21–26, August 2011.

[34] Lijun Qian and Ning Song. Detecting and locating wormhole attacks in wireless ad hoc networks through statistical analysis of multi-path. In *Wireless Communications and Networking Conference, 4:2106 âĂŞ 2111*, 2005.

[35] Jonny Karlsson, Laurence S. Dooley, and GÃűran Pulkkis. A new MANET wormhole detection algorithm based on traversal time and hop count analysis. *Sensors*, 11(12):11122–11140, 2011.

[36] Bruce Schneier. *Applied Cryptography (2nd Ed.): Protocols, Algorithms, and Source Code in C*. John Wiley & Sons, Inc., New York, NY, USA, 1995.

[37] Joseph Migga Kizza. *Guide to Computer Network Security*. Springer-Verlag, London, UK, 2010.

[38] D. Coppersmith, D. B. Johnson, S. M. Matyas, T. J. Watson, Don B. Johnson, and Stephen M. Matyas. *Triple DES Cipher Block Chaining with Output Feedback Masking*. Research report. IBM T.J. Watson Research Center, 1996.

[39] Frederic P. Miller, Agnes F. Vandome, and John McBrewster. *Advanced Encryption Standard*. Alpha Press, 2009.

[40] Ronald L. Rivest. The RC5 encryption algorithm. In *Proceedings 2nd International Workshop on Fast Software Encryption*, volume 1008 of *Lecture Notes in Computer Science*, pages 86–96, Leuven, Belgium, 1994. Springer.

[41] Alexander W. Dent. Choosing key sizes for cryptography. *Information Security Technical Report*, 15(1):21–27, February 2010.

[42] R.L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21:120–126, 1978.

[43] Darrel Hankerson, Alfred J. Menezes, and Scott Vanstone. *Guide to Elliptic Curve Cryptography*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003.

[44] Adam Shostack. *Threat Modeling: Designing for Security*. John Wiley & Sons, Inc., New York, NY, USA, 2014.

[45] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Keying hash functions for message authentication. In *Proceedings of the 16th Annual International Cryp-*

tology Conference on Advances in Cryptology, CRYPTO '96, pages 1–15, London, UK, UK, 1996. Springer-Verlag.

[46] Ronald L. Rivest. The MD5 message-digest algorithm. Internet RFC 1321, April 1992.

[47] D. Eastlake and P. Jones. US Secure Hash Algorithm 1 (SHA1). RFC 3174, United States, 9 2001.

[48] CORPORATE NIST. The digital signature standard. *Commun. ACM*, 35(7):36–40, July 1992.

[49] Kimaya Sanzgiri, Daniel Laflamme, Bridget Dahill, Brian Neil Levine, Clay Shields, and Elizabeth M. Belding-royer. Authenticated routing for ad hoc networks. *IEEE Journal On Selected Areas In Communications*, 23:598–610, 2005.

[50] Seung Yi, Prasad Naldurg, and Robin Kravets. Security-aware ad hoc routing for wireless networks. In *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '01, pages 299–302, New York, NY, USA, 2001. ACM.

[51] B. Dahill, K. Sanzgiri, B. N. Levine, E. M. Belding-Royer, and C. Shields. A secure routing protocol for ad hoc networks. Technical report, University of Massachusetts, USA, 2002.

[52] Yih-Chun Hu, David B. Johnson, and Adrian Perrig. SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks. In *Proceedings of the 4th IEEE Workshop on Mobile Computing Systems and Applications*, WMCSA'02, pages 3–13, Callicoon, NY, USA, 2002. IEEE.

[53] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. ARIANDE: a secure on-demand routing protocol for ad hoc networks. *Wirel. Netw.*, 11(1-2):21–38, January 2005.

[54] Manel Guerrero Zapata. Secure ad hoc on-demand distance vector routing. *SIGMOBILE Mob. Comput. Commun. Rev.*, 6(3):106–107, June 2002.

[55] Panagiotis Papadimitratos and Zygmunt J. Haas. Secure link state routing for mobile ad hoc networks. In *Symposium on Applications and the Internet Workshops*, pages 379–383. IEEE Computer Society, 2003.

[56] Baruch Awerbuch, Reza Curtmola, David Holmer, Cristina Nita-Rotaru, and Herbert Rubens. Odsbr: An on-demand secure byzantine resilient routing protocol for wireless ad hoc networks. *ACM Trans. Inf. Syst. Secur.*, 10(4):6:1–6:35, January 2008.

[57] Sonja Buchegger and Jean-Yves Le Boudec. Performance analysis of the confidant protocol. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '02, pages 226–236, New York, NY, USA, 2002. ACM.

[58] Minji Kim, Muriel MÃľdard, JoÃčo Barros, and Ralf KÃűtter. An algebraic watchdog for wireless network coding. In *Proceedings of IEEE ISIT*, 2009.

[59] A. Bandyopadhyay, S. Vuppala, and P. Choudhury. A simulation analysis of flooding attack in MANET using ns-3. In *Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology (Wireless VITAE), 2011 2nd International Conference on*, pages 1–5, 2011.

[60] Nidhi Sharma and Alok Sharma. The black-hole node attack in MANET. In *Proceedings of the 2012 Second International Conference on Advanced Computing & Communication Technologies*, ACCT '12, pages 546–550, Washington, DC, USA, 2012. IEEE Computer Society.

[61] Wan Du, David Navarro, Fabien Mieyeville, and Frédéric Gaffiot. Towards a taxonomy of simulation tools for wireless sensor networks. In *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*, SIMUTools '10, pages 1–7, Brussels, Belgium, Belgium, 2010. ICST (Institute

for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[62] Teerawat Issariyakul and Ekram Hossain. *Introduction to Network Simulator NS2*. Springer Publishing Company, 2 edition, 2012.

[63] Raj Jain. The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling. *SIGMETRICS Performance Evaluation Review*, 19(2):5–11, September 1991.

[64] Ahmad Ali Abdullah, Fayez Gebali, and Lin Cai. Modeling the throughput and delay in wireless multihop ad hoc networks. In *IEEE Conference on Global Telecommunications (GLOBECOM)*, pages 1–6. IEEE, 2009.

[65] F. Liu, C. Lin, H. Wen, and P. Ungsunan. Throughput analysis of wireless multi-hop chain networks. In *Eighth IEEE/ACIS International Conference on Computer and Information Science (ICIS)*, pages 834–839, June 2009.

[66] Qualnet communications simulation. `http://web.scalable-networks.com/content/qualnet` [retrieved: April, 2016].

[67] Opnet. `http://www.riverbed.com/products/steelcentral/opnet.html` [retrieved: April, 2016].

[68] Matlab. `http://www.mathworks.com/products/matlab` [retrieved: April, 2016].

[69] The network simulator ns-2. `http://www.isi.edu/nsnam/ns/` [retrieved: April, 2016].

[70] Omnet++. `https://omnetpp.org/` [retrieved: April, 2016].

[71] Glomosim. `http://pcl.cs.ucla.edu/projects/glomosim/` [retrieved: April, 2016].

[72] W. Kasch, J. Ward, and J. Andrusenko. Wireless network modeling and simulation tools for designers and developers. *Communications Magazine, IEEE*, 47(3):120–127, March 2009.

[73] Tracy Camp, Jeff Boleng, and Vanessa Davies. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Ccomputing (WCMC): Special Issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2:483–502, 2002.

[74] Christian Schindelhauer. Mobility in wireless networks. In *In 32nd Annual Conference on Current Trends in Theory and Practice of Informatics, Czech*, 2006.

[75] Jon Viega, Pravir Chandra, and Matt Messier. *Network Security with Openssl*. O'Reilly & Associates, Inc., Sebastopol, CA, USA, 1st edition, 2002.

[76] Mohamed A. Abdelshafy and Peter J.B. King. AODV & SAODV under attack: Performance comparison. In *ADHOC-NOW 2014, LNCS 8487*, pages 318–331, Benidorm, Spain, Jun 2014.

[77] Mohamed A. Abdelshafy and Peter J.B. King. Analysis of security attacks on AODV routing. In *8th International Conference for Internet Technology and Secured Transactions (ICITST)*, pages 290–295, London, UK, Dec 2013.

[78] Mohamed A. Abdelshafy and Peter J.B. King. AODV routing protocol performance analysis under MANET attacks. *International Journal for Information Security Research (IJISR)*, 3(1/2):418–426, March/June 2013.

[79] Mohamed A. Abdelshafy and Peter J.B. King. Dynamic source routing under attacks. In *7th International Workshop on Reliable Networks Design and Modelling (RNDM)*, pages 174–180, Munich, Germany, Oct 2015.

[80] Ping Yi, Zhoulin Dai, Yi-Ping Zhong, and Shiyong Zhang. Resisting flooding attacks in ad hoc networks. In *International Conference on Information Technology: Coding and Computing (ITCC)*, volume 2, pages 657–662, April 2005.

[81] Bo-Cang Peng and Chiu-Kuo Liang. Prevention techniques for flooding attacks in ad hoc networks. In *3rd Workshop on Grid Technologies and Applications (WoGTA 06)*, pages 657–662 Vol. 2, Hsinchu, Taiwan, December 2006.

[82] Jian-Hua Song, Fan Hong, and Yu Zhang. Effective filtering scheme against RREQ flooding attack in mobile ad hoc networks. In *Proceedings of the Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies*, PDCAT '06, pages 497–502, Washington, DC, USA, 2006. IEEE Computer Society.

[83] V. Balakrishnan, V. Varadharajan, U. Tupakula, and M.E.G. Moe. Mitigating flooding attacks in mobile ad-hoc networks supporting anonymous communications. In *2nd International Conference on Wireless Broadband and Ultra Wideband Communications (AusWireless)*, pages 29–34, Aug 2007.

[84] Revathi Venkataraman, M. Pushpalatha, Rishav Khemka, and T. Rama Rao. Prevention of flooding attacks in mobile ad hoc networks. In *Proceedings of the International Conference on Advances in Computing, Communication and Control*, ICAC3 '09, pages 525–529, New York, NY, USA, 2009. ACM.

[85] Ujwala D. Khartad and R. K. Krishna. Route request flooding attack using trust based security scheme in MANET. *International Journal of Smart Sensors and Ad Hoc Networks (IJSSAN)*, 1(4):27–33, 2012.

[86] Mohamed A. Abdelshafy and Peter J.B. King. Resisting flooding attacks on AODV. In *8th International Conference on Emerging Security Information, Systems and Technologies (SECURWARE)*, pages 14–19, Lisbon, Portugal, Nov 2014.

[87] Mohamed A. Abdelshafy and Peter J.B. King. Flooding attacks resisting mechanism in MANETs. Submitted to IEEE International Conference on Communications (ICC) 2016.

[88] Mohamed A. Abdelshafy and Peter J.B. King. Resisting blackhole attacks on MANETs. In *13th IEEE Consumer Communications and Networking Conference (CCNC)*, pages 1–6, Las Vegas, USA, Jan 2016.

[89] Seungjoon Lee, Bohyung Han, and Minho Shin. Robust routing in wireless ad

hoc networks. In *International Conference on Parallel Processing Workshops*, pages 73–78, 2002.

[90] Satoshi Kurosawa, Hidehisa Nakayama, Nei Kato, Abbas Jamalipour, and Yoshiaki Nemoto. Detecting blackhole attack on AODV-based mobile ad hoc networks by dynamic learning method. *International Journal of Network Security*, 5(3):338–346, 2007.

[91] L. Tamilselvan and V. Sankaranarayanan. Prevention of blackhole attack in MANET. In *2nd International Conference on Wireless Broadband and Ultra Wideband Communications*, pages 21–21, Aug 2007.

[92] Payal N. Raj and Prashant B. Swadas. DPRAODV: a dynamic learning system against blackhole attack in AODV based MANET. *International Journal of Computer Science Issues (IJCSI)*, 2:54–59, 2009.

[93] Nital Mistry, Devesh C Jinwala, and Mukesh Zaveri. Improving AODV protocol against blackhole attacks. In *International MultiConference of Engineers and Computer Scientists (IMECS)*, pages 1–5, Hong Kong, China, March 2010.

[94] Xin Li, Zhiping Jia, Peng Zhang, and Haiyang Wang. A trust-based multipath routing framework for mobile ad hoc networks. In *7th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, volume 2, pages 773–777, Aug 2010.

[95] Rajdipsinh Vaghela, Divyesh Yoganand, and Monika Changela. A survey on approaches towards the black hole attack in MANET. *Indian Journal of Research*, 1(12):58–61, 2012.

[96] Tarek M. Mahmoud, Abdelmgeid A. Aly, and Omar Makram. A modified AODV routing protocol to avoid black hole attack in MANETs. *International Journal of Computer Applications*, 109(6):27–33, January 2015.

[97] N. Choudhary and L. Tharani. Preventing black hole attack in AODV using timer-based detection mechanism. In *International Conference on Signal Pro-*

cessing And Communication Engineering Systems (SPACES), pages 1–4, Jan 2015.

[98] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (AODV) routing. Internet Draft RFC 3561 (Experimental), July 2003.